

# UNIVERSIDAD POLITÉCNICA DE MADRID



ESCUELA UNIVERSITARIA DE  
INGENIERÍA  
TÉCNICA DE TELECOMUNICACIÓN



PROYECTO FIN DE CARRERA

## **ANDROID EN EL HOGAR DIGITAL USANDO UPnP Y REDES LONWORKS**

Roberto Augusto Cuenca Cuenca.

Mayo – 2013



## PROYECTO FIN DE CARRERA PLAN 2000

**TEMA:**

**TITULO:** Android en el Hogar Digital usando UPnP y Redes LonWorks

**AUTOR:** Roberto Augusto Cuenca Cuenca

**TUTOR:** Rubén de Diego Martínez

**DEPARTAMENTO:** DIATEL

**MIEMBROS DEL TRIBUNAL CALIFICADOR:**

**PRESIDENTE:** Miguel Ángel Freire Rubio

**VOCAL:** Rubén de Diego Martínez

**VOCAL SECRETARIO:** Miguel Ángel Valero Duboy

**Fecha de lectura:**

**Calificación:**

**El Secretario,**

**RESUMEN DEL PROYECTO:**

Este Proyecto Final de Carrera se centra principalmente en el estudio de las tecnologías aplicadas al Hogar Digital (HD).

La primera tecnología estudiada es LonWorks, escogida al contarse con un entorno domótico de este tipo en la EUITT, materializado en una maqueta de un HD.

La segunda tecnología es Universal Plug and Play (UPnP), que se ha usado para recubrir la tecnología LonWorks de forma que los dispositivos LonWorks instalados en la maqueta puedan utilizarse como si fueran dispositivos UPnP. Para conseguir realizar el recubrimiento se han utilizado las tecnologías de servicios web, SOAP y XML.

Finalmente se ha desarrollado una aplicación que se ejecuta en terminales, SmartPhones o Tablets, Android con la que se realiza el control de la maqueta del Hogar Digital LonWorks.



# AGRADECIMIENTOS

---

Quiero agradecer principalmente a mis padres por todo el esfuerzo que han hecho a lo largo de estos años, dándome la oportunidad de estudiar y ser quien soy. No hay palabras para expresar mi eterno agradecimiento.

A mis hermanos.

A mi familia por todo el apoyo prestado a lo largo de estos años. En especial a mi abuela, que donde quiera que estés te agradezco todo el cariño que me has dado.

A mi tutor Rubén por la confianza depositada en mí.

A todos mis amigos y conocidos que a lo largo de estos años me han apoyado en los buenos y malos momentos. En especial a mis amigos de fatigas, Sergio (el Jefe), Sergio (Aroca), Javi (Pipi), Lujan, Edu, David (Vasco), Santi, Helena, Noelia, Fer, Oscar y aunque no os nombre, siempre llevare conmigo un poquito de todos vosotros.

Lunes blancos...



# RESUMEN

---

Este Proyecto Final de Carrera se centra principalmente en el estudio de las tecnologías aplicadas al Hogar Digital (HD) y así poder desarrollar nuevas herramientas aplicadas a este Sector.

La primera tecnología estudiada es LonWorks que se escogió porque en la EUITT se cuenta con un entorno domótico donde se aplica esta tecnología, materializado como una maqueta de una instalación LonWorks en un HD. Sobre este entorno se ha desarrollado una nueva forma de gestionar y controlar los elementos de una Red LonWorks mediante la tecnología Universal Plug & Play (UPnP).

Los dispositivos LonWorks han sido recubiertos con una capa con la que se consigue tratar los elementos LonWorks como dispositivos UPnP, pudiendo de esta manera trabajar bajo un mismo formato de dispositivo. Este formato está definido por un documento denominado *Descripción del Dispositivo* UPnP. Por tanto, no es necesario conocer el estado de los dispositivos de la Red LonWorks, sino únicamente trabajar bajo una Red UPnP, creando nuevos dispositivos UPnP, puntos de Control y servicios a partir de los elementos LonWorks.

Una vez realizado el recubrimiento del sistema LonWorks con UPnP, se ha definido y desarrollado una aplicación para Android que, permite controlar los elementos de la maqueta del HD desde un *Smartphone* o una *Tablet*. El acceso a la Red LonWorks de la maqueta del HD se hace a través de la interfaz Web Services SOAP/XML del dispositivo iLON100.

## *Palabras Clave*

*Hogar Digital, Domótica, LonWorks, SOAP, servicio web, XML, UPnP, Android, Cling.*



# ABSTRACT

---

This Final Degree Project is mainly focused on the study of the technologies applied to Digital Home and thus be able to develop new tools applied to this area.

The first study is LonWorks technology which was chosen because the EUITT has a domotic environment where this technology is applied, materialized as a demonstrator of a LonWorks installation in HD. With this environment has developed a new way to manage and control the elements of a LonWorks network through technology by Universal Plug & Play (UPnP).

LonWorks devices are covered with a layer and it is achieved LonWorks elements treat as UPnP devices. In this way the LonWorks elements can work under one device format. This format is defined by a document called UPnP Device Description. Therefore, it isn't necessary to know the state of the LonWorks network devices, but only work under an UPnP network, creating new UPnP devices, control points and services from LonWorks elements.

After backfill of LonWorks system with UPnP, have being defined and developed an Android application that allow controlling the elements of the mockup of HD from Smartphone or a Tablet. The LonWorks network Access of the mockup of HD is done through the Web Interface Services SOAP / XML iLON100 device.

## *Keywords*

*Hogar Digital, Domótica, LonWorks, SOAP, servicio web, XML, UPnP, Android, Cling.*





# ÍNDICE

AGRADECIMIENTOS.....	i
RESUMEN .....	iii
ABSTRACT .....	v
Índice de Figuras .....	e
Índice de Tablas.....	g
1. Presentación del proyecto .....	1
1.1. Introducción .....	1
1.2. Objetivos del Proyecto Fin de Carrera .....	2
1.3. Organización de la memoria .....	3
2. Introducción al Hogar Digital.....	5
2.1. Introducción .....	5
2.2. Hogar Digital.....	5
2.2.1. Domótica .....	5
2.3. Paneles demostrativos .....	6
3. Análisis de las Tecnologías utilizadas .....	9
3.1. UPnP.....	9
3.1.1. Red UPnP .....	9
3.1.1.1. Dispositivos .....	9
3.1.2. Cling.....	11
3.1.2.1. Cling Core .....	11
3.2. LonWorks .....	12
3.2.1. LonTalk .....	12
3.2.2. iLON100.....	13
3.2.2.1. Especificaciones funcionales del iLon100 Internet Server .....	14
3.2.3. Lonmaker.....	14
3.3. Android.....	15
3.3.1. ¿Qué es Android? .....	15
3.3.1.1. Breve Historia .....	15
3.3.2. Principales Características.....	16
3.3.3. Evolución .....	17
3.3.4. Arquitectura .....	18
3.3.5. Ciclo de vida de una Aplicación .....	20

4.	Antecedentes y Análisis del Sistema .....	21
4.1.	Experiencias Relacionadas .....	21
4.2.	Análisis de especificaciones .....	22
4.2.1.	Requisitos funcionales.....	22
4.2.1.1.	Diagramas de Casos de Uso .....	22
5.	Diseño del Sistema .....	27
5.1.	Entorno.....	27
5.2.	Servicios UPnP .....	29
5.3.	Configuración de la Red LonWorks con la herramienta de diseño Lonmaker. ....	30
5.4.	Integración de elementos LonWorks como Dispositivos UPnP .....	35
5.5.	Control de Dispositivos UPnP .....	37
5.6.	Diseño de Componentes Software.....	38
5.6.1.	Diagramas de Secuencia.....	38
5.6.1.1.	Cargar UPnP Stack .....	39
5.6.1.2.	Buscar Elementos LonWorks y Crear Dispositivos UPnP.....	39
5.6.1.3.	Visualizar Área con Dispositivos UPnP .....	40
5.6.1.4.	Invocar Acciones de un Dispositivo UPnP .....	41
5.6.1.5.	Recoger Eventos generados por las Acciones de un dispositivo UPnP .....	42
5.7.	Diagramas de Paquetes y Clases .....	43
6.	Pruebas y validación.....	45
6.1.	Pruebas de la Aplicación Android con los Paneles Demostrativos .....	45
7.	Conclusiones .....	49
7.1.	Trabajos Futuros.....	50
7.1.1.	Servicios Dinámicos de un Dispositivo UPnP .....	50
7.1.2.	Mejorar la vista de la Aplicación Android .....	50
7.1.3.	Acceso a la Aplicación .....	50
7.1.4.	Compatibilidad de Dispositivos no UPnP .....	51
8.	Anexo A: Gestión del Proyecto.....	53
8.1.	Fichero WSDL Versión 1.1 .....	53
8.2.	Android Manifest .....	55
9.	Anexo B: Manual del desarrollador.....	57
9.1.	Instalación del Sistema.....	57
9.2.	Interfaces de la Aplicación .....	57
10.	Anexo C: Conexionado detallado de Red LonWorks .....	61

10.1.	Nodo de Control Estándar INS-231 .....	61
10.2.	Nodo Regulador Analógico INS-360 .....	63
10.3.	Configuración bloque funcional ILON.SNVT_switch .....	65
11.	Anexo D: Manual de Usuario .....	67
11.1.	Inicio de los Servicios de la UPnP Stack.....	67
11.2.	Búsqueda de elementos LonWorks y creación de nuevos dispositivos UPnP. ....	68
11.3.	Control y Visualización de Dispositivo UPnP .....	69
12.	Acrónimos .....	73
13.	Bibliografía .....	75



# Índice de Figuras

---

Figura 1 : Paneles demostrativos LonWorks .....	6
Figura 2 : Plano de interconexión de dispositivos.....	8
Figura 3 : Descripción Dispositivo UPnP.....	10
Figura 4 : Lontalk Architecture .....	13
Figura 5 : Dispositivo iLON100 Internet Server .....	14
Figura 6 : Evolución del Sistema Android.....	17
Figura 7 : Arquitectura Android .....	19
Figura 8 : Ciclo de Vida de una Aplicación.....	20
Figura 9 : Diagrama de Casos de Uso Principal .....	23
Figura 10: Diagrama de casos de uso Inicializar Sistema .....	24
Figura 11: Diagramas de Casos de Uso - Control de Dispositivo UPnP .....	25
Figura 12 : Escenario Físico .....	28
Figura 13: Arquitectura Sistema.....	29
Figura 14 : Lonmaker Network Design .....	30
Figura 15 : Lonmaker Network Design - 2 .....	31
Figura 16 : New Device Wizard .....	31
Figura 17 : Configuración Cuadro Domótico .....	33
Figura 18 : Configuración Dispositivos de las Áreas (Entrada, Salón, Baño, Cocina y Dormitorio). .....	34
Figura 19 : Formato fichero home_config.xml.....	36
Figura 20 : Diagrama de Secuencia - Cargar Pila UPnP .....	39
Figura 21 : Diagrama de Secuencia - Dispositivo UPnP desde elemento LonWorks.....	39
Figura 22 : Diagrama de Secuencia - Área con Dispositivos UPnP .....	40
Figura 23 : Diagrama de Secuencia - Invocar Acciones .....	41
Figura 24 : Diagrama de Secuencia - Captura de Eventos de un Dispositivo UPnP .....	42
Figura 25 : Diagrama de Paquetes .....	43
Figura 26 : Diagrama de Clases upm.ro.hogar .....	44
Figura 27 : Cargando Dispositivos en Tablet con Sistema Operativo Android versión 2.2 .....	46
Figura 28 : Comparación del Área Cocina en Aplicación Android y en los Paneles Demostrativos. .....	46
Figura 29 : Prueba de Regulación de Luz.....	47
Figura 30 : Prueba de Detención de Sensor de Presencia.....	47
Figura 31 : Fichero WSDL parte 1 .....	53
Figura 32 : Fichero WSDL parte 2 .....	54
Figura 33 : Android Manifest.....	55
Figura 34 : Interfaz BuildDeviceLonWorks .....	58
Figura 35 : Interfaz TypesDevices .....	59
Figura 36 : Interfaz TypeUPnP.....	60
Figura 37 : Layout dormitorio.xml.....	60
Figura 38 : Nodo de Control INS-231 .....	61
Figura 39: Nodo de Control INS-360 .....	64
Figura 40 : Cargar Servicios UPnP.....	67

Figura 41 : Primera vista..... 68

Figura 42 : Buscando elementos LonWorks ..... 68

Figura 43 : Segunda Vista ..... 69

Figura 44 : Tercera Vista..... 70

Figura 45 : Proceso inicial de Carga de Estados de los Dispositivos UPnP ..... 70

Figura 46 : Dialogo para Regular Luz del Salón. .... 71

# Índice de Tablas

---

Tabla 1 : Variables utilizadas del INS-231.....	62
Tabla 2 : Relación de conexiones con el Nodo de Control estándar NS-231 .....	63
Tabla 3 : Conexiones BF iLon.SNVT_switch del Dispositivo iLON100.....	65





# 1. Presentación del proyecto

---

## 1.1. Introducción

El proyecto “*Android en el Hogar Digital usando UPnP y Redes LonWorks*”, nació de la intención de conocer y tratar los temas relacionados con el Hogar Digital usando el Sistema Operativo Android. De aquí, surge la idea de trabajar con las tecnologías UPnP y LonWorks.

La Escuela Universitaria de Ingeniería Técnica de Telecomunicación (EUITT) de la Universidad Politécnica de Madrid (UPM) cuenta con un entorno domótico [1] dónde se aplica la tecnología LonWorks, en adelante denominado *Paneles Demostrativos*, que es una maqueta de una instalación LonWorks en un Hogar Digital.

Dada la creciente importancia de la tecnología UPnP, base de la norma DLNA [2] (Digital Living Network Alliance) se decidió por estudiar la posibilidad de integrar ambas tecnologías desarrollando un recubrimiento UPnP para LonWorks.

Gracias a la ayuda del uso de la tecnología UPnP este entorno contará con una nueva forma de ver los dispositivos de la Red LonWorks, además de contar así, con todas las ventajas que ofrece la tecnología UPnP proporcionada con la ayuda de las librerías y herramientas Cling en entornos de desarrollo Android.

Realizado el recubrimiento UPnP de los Paneles Demostrativos, se ha desarrollado una aplicación Android, capaz de correr en SmartPhones y Tablets, que permite controlar todos los elementos LonWorks existentes en la maqueta.

## 1.2. Objetivos del Proyecto Fin de Carrera

La idea fundamental a llevar a cabo en este Proyecto Final de Carrera (PFC) es la de crear una aplicación Android capaz de contener todas las funcionalidades de la *UPnP Stack*, que permita controlar todos los elementos existentes de los dispositivos LonWorks de la red instalada en los Paneles Demostrativos.

Los objetivos parciales de este PFC son los siguientes:

- ▶ Realizar un estudio de las tecnologías y del entorno domótico utilizado.
- ▶ Enumerar los nuevos servicios y funcionalidades que debe tener el entorno.
- ▶ Diseñar los mecanismos necesarios para la integración de la tecnología UPnP y la tecnología LonWorks.
- ▶ Diseñar la aplicación Android para el modelado, visualización y control de los elementos del entorno domótico utilizado, aplicando los nuevos servicios y funcionalidades resultantes.
- ▶ Implementar y construir los mecanismos resultantes tras el diseño de la integración de las tecnologías usadas en la aplicación Android.

### 1.3. Organización de la memoria

La organización de la memoria llevada a cabo intenta seguir en todo momento un sentido lógico, citando los apartados a los que se hace relación y definiendo los conceptos que serán usados en capítulos posteriores.

En primer lugar, se presenta este proyecto con este primer capítulo, el cual consta de la introducción, los objetivos y la organización de la memoria.

A continuación con los capítulos 2 y 3 se realiza el estudio del entorno domótico y de las tecnologías utilizadas respectivamente.

En el capítulo 4 se realiza el análisis del sistema y se definen las características que debe tener. El termino *sistema* se refiere al conjunto de funcionalidades y servicios aplicados al entorno domótico, basados en las tecnologías UPnP y LonWorks que son integrados bajo la aplicación Android a construir.

En el capítulo 5 se estudia el Diseño del Sistema. En la primera parte se analiza la arquitectura del sistema junto con los mecanismos empleados en la solución. En la segunda parte de este capítulo se estudian los componentes del mismo con la ayuda de los correspondientes diagramas UML (*Unified Modeling Language*), todo ello contemplado desde el punto de vista de la implementación de la aplicación Android.

En el capítulo 7 se describen las conclusiones a las que se ha llegado, junto con las propuestas que se plantean a futuro para posibles líneas de trabajo a seguir a partir de la realización de este PFC.

Por último, la memoria consta de varios Anexos y de la bibliografía utilizada.



## 2. Introducción al Hogar Digital

---

### 2.1. Introducción

El constante cambio de la tecnología en todos los ámbitos provoca que continuamente se desarrollen nuevas herramientas para la gestión y desarrollo que sean más eficientes, más económicas, etc. Con la realización de este PFC se ha buscado dar, una nueva herramienta para gestionar un Hogar Digital basado en el estándar LonWorks.

En este capítulo se realiza un estudio del Hogar Digital utilizado para el desarrollo y verificación del sistema a construir en este PFC. El termino *sistema* se utilizara más adelante y será entendido como el conjunto de funcionalidades y servicios aplicados al entorno domótico, basados en las tecnologías UPnP y LonWorks que son integrados bajo la aplicación Android a construir.

### 2.2. Hogar Digital

Hogar Digital [3] es un término utilizado hoy en día para definir a los equipos y sistemas que se integran para ofrecer funciones y servicios que facilitan la gestión y el mantenimiento del hogar. El conjunto de sistemas capaces de automatizar una vivienda se conoce con el termino Domótica.

La Domótica ofrece servicios que mejoraran la eficiencia y la gestión en un hogar estás mejoras son aplicadas a los recursos energéticos, comunicación, seguridad del hogar. Además, estas mejoras, suelen traer consigo una reducción de costes y tiempos como en la puesta en marcha de las mejoras en el Hogar.

#### 2.2.1. Domótica

La domótica [4], es un término con el que definen los automatismos y la robótica en el hogar. En concreto, la domótica es la rama tecnológica, que se dedica a investigar

y desarrollar las aplicaciones de las Tecnologías de la información y comunicación (TIC) en el hogar.

El objetivo no es otro que facilitar la gestión que se lleva a cabo cuando se realizan tareas en la vivienda. Existen varios estándares que se utilizan en este contexto entre los cuales cabe destacar *LonWorks*, *Konnex*, *X10*, *OSGI*, *UPnP* y *ZigBee*.

### 2.3. Paneles demostrativos

El escenario físico que se va a utilizar para llevar a cabo las verificaciones del cumplimiento de los objetivos, es el implementado en los *Paneles Demostrativos* [1] ilustrado en la Figura 1, que proporciona el entorno domótico basado en el estándar *LonWorks*.

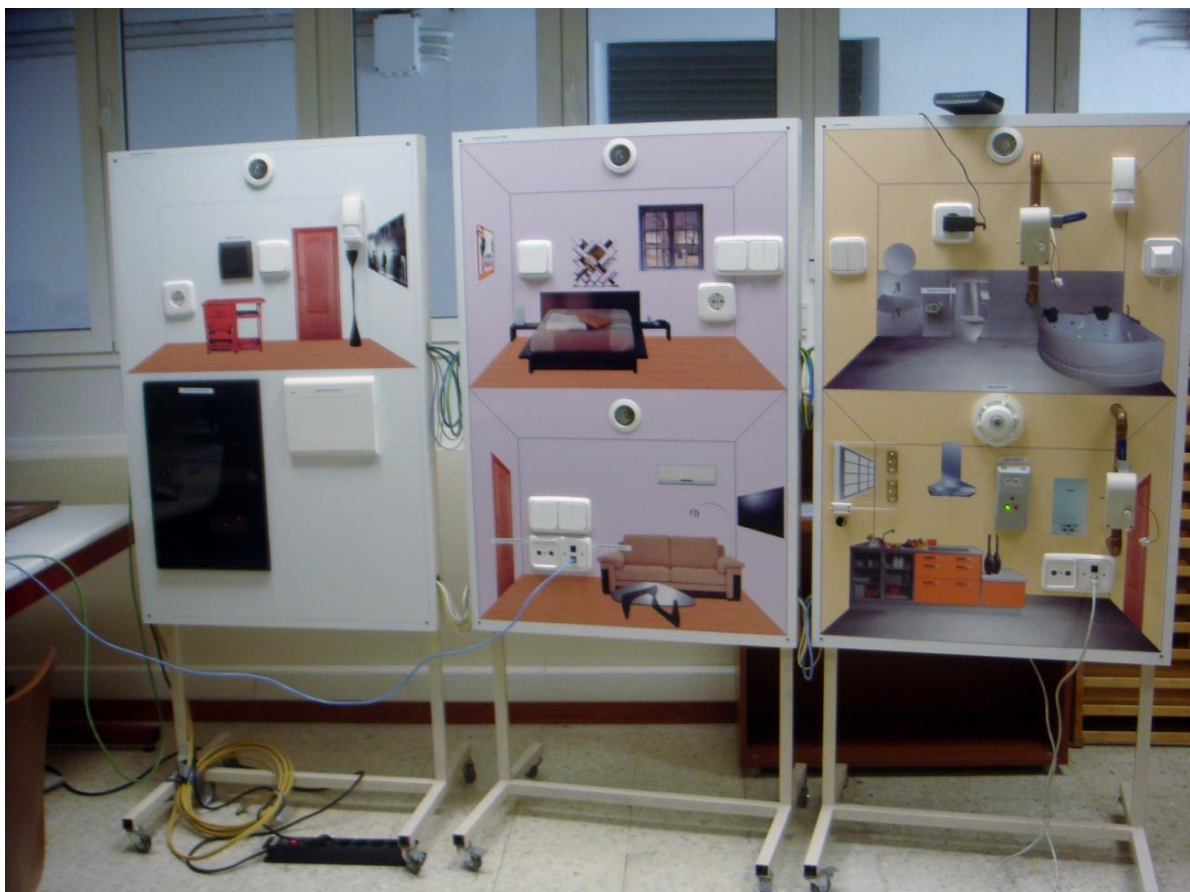


Figura 1 : Paneles demostrativos LonWorks

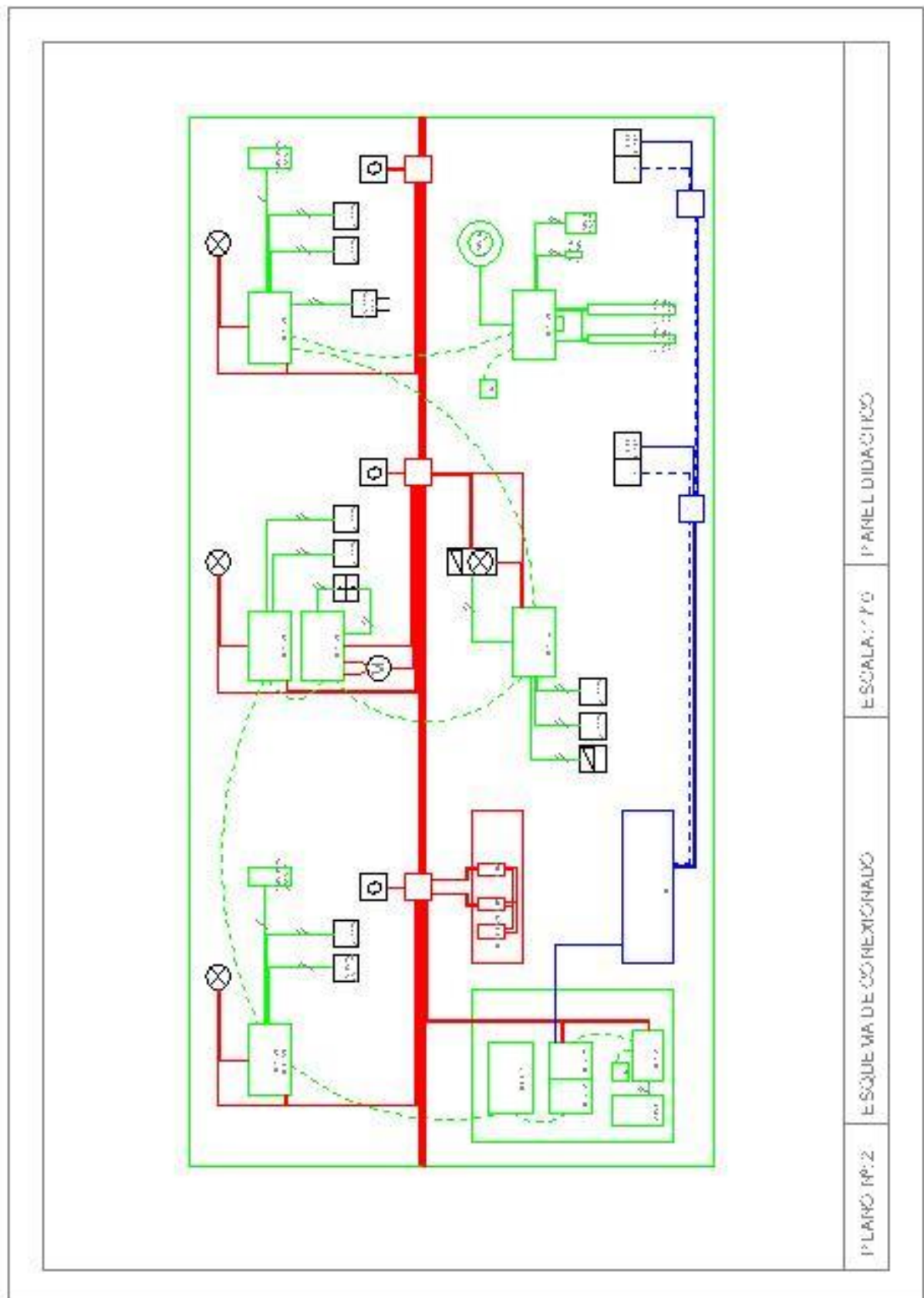
La estructura de estos paneles demostrativos se divide en seis zonas o áreas.

Las zonas representan las distintas partes de un hogar como son la entrada, el dormitorio, el baño, el salón y la cocina de un hogar. Salvo la existencia de una zona denominada cuadro domótico que alberga los dispositivos INM-011, INM-020, IFA-200 y *ILon100 Smart Server*, esta zona es utilizada para el control y supervisión de los elementos del Hogar (luz, sensores, actuadores, etc.) con la ayuda de los dispositivos anteriormente mencionados. Es importante resaltar al dispositivo iLon100 [5], que es un servidor de aplicaciones que se utiliza como pasarela entre la aplicación desarrollada y el mundo LonWorks.

En el resto de las zonas se hace uso del dispositivo INS-231 excepto en la zona del salón donde se utiliza el dispositivo INS-360. Se puede encontrar más información sobre los dispositivos INS-231 y INS-360 en el Anexo C.

En la Figura 2 se representa la interconexión de los dispositivos en los Paneles demostrativos.





**Figura 2 : Plano de interconexión de dispositivos**

## 3. Análisis de las Tecnologías utilizadas

---

En este capítulo se realiza un estudio de las tecnologías utilizadas en este PFC indicando qué son y para qué son usadas.

### 3.1. UPnP

La tecnología UPnP [6] es promovida por el *UPnP Forum* que fue fundado en Octubre de 1999, siendo una organización formada por más de 986 compañías pertenecientes a la gran mayoría de los sectores de la tecnología.

Una de las ventajas por la que ha tenido tanto éxito la tecnología UPnP es porque tanto sus dispositivos como los servicios que ofrecen están definidos en *Documentos XML*. Además hacen uso de los protocolos *TCP/IP*.

Dentro de sus características se pueden destacar las siguientes:

- ◆ Es una arquitectura abierta y distribuida.
- ◆ Está basada en las tecnologías base de internet como son TCP/IP.
- ◆ Es independiente de un lenguaje de programación o de un Sistema Operativo. Por tanto lo hace que sea muy fácil integrarlo con otras tecnologías.

#### 3.1.1. Red UPnP

Una Red UPnP lo forman: los dispositivos, servicios y los puntos de control.

##### 3.1.1.1. Dispositivos

Un Dispositivo UPnP contiene servicios que son ofrecidos a otros dispositivos de la red. Toda la información que contiene un dispositivo está recogida en el

*documento XML descripción del dispositivo*, que se almacena en el propio dispositivo.

El Documento XML de *descripción del dispositivo* tiene el siguiente formato:

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0" configId="configuration number">
  <specVersion>
    <major>1</major>
    <minor>1</minor>
  </specVersion>
  <device>
    <deviceType>urn:schemas-upnp-org:device:deviceType:v</deviceType>
    <friendlyName>short user-friendly title</friendlyName>
    <manufacturer>manufacturer name</manufacturer>
    <manufacturerURL>URL to manufacturer site</manufacturerURL>
    <modelDescription>long user-friendly title</modelDescription>
    <modelName> model name </modelName>
    <modelNumber>model number</modelNumber>
    <modelURL>URL to model site </modelURL>
    <serialNumber>manufacturer's serial number</serialNumber>
    <UDN>uuid:UUID</UDN>
    <UPC>Universal Product Code</UPC>
    <iconList>
      <icon>
        <mimetype>image/format</mimetype>
        <width>horizontal pixels</width>
        <height>vertical pixels</height>
        <depth>color depth</depth>
        <url>URL to icon</url>
      </icon>
      <!-- XML to declare other icons, if any, go here -->
    </iconList>
    <serviceList>
      <service>
        <serviceType>urn:schemas-upnp-org:service:serviceType:v
        </serviceType>
        <serviceId>urn:upnp-org:serviceId:serviceID
        </serviceId>
        <SCPURL>URL to service description</SCPURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
      </service>
      <!-- Declarations for other services defined by a UPnP Forum working committee
      (if any) go here -->
      <!-- Declarations for other services added by UPnP vendor (if any) go
      here -->
    </serviceList>
    <deviceList>
      <!-- Description of embedded devices defined by a UPnP Forum working committee
      (if any) go here -->
      <!-- Description of embedded devices added by UPnP vendor (if any) go
      here -->
    </deviceList>
    <presentationURL>URL for presentation</presentationURL>
  </device>
</root>
```

FUENTE: [HTTP://UPNP.ORG/SPECS/ARCH/UPNP-ARCH-DEVICEARCHITECTURE-V1.0.PDF](http://UPNP.ORG/SPECS/ARCH/UPNP-ARCH-DEVICEARCHITECTURE-V1.0.PDF)

**Figura 3 : Descripción Dispositivo UPnP**

### 3.1.2. Cling

Cling [7] es un entorno JAVA compatible con la UPnP Stack y el Cling Core es compatible con el *UPnP Device Architecture* 1.0. Su uso es libre y es distribuido bajo los términos de la *GNU Lesser General Public License*.

#### 3.1.2.1. Cling Core

Cling Core son librerías UPnP JAVA que implementan la infraestructura que ofrece los servicios necesarios para trabajar con la tecnología UPnP.

Las principales características [8] de estas librerías son:

- ◆ Poseen interfaces de programación de aplicaciones seguras (API)

Se ofrece una serie de interfaces de programación para los desarrolladores de aplicaciones. Estas interfaces proporcionan cada cabecera UPnP, mensaje y metainformación como un tipo seguro teniendo pocos tipos de casting, facilitando así un entorno en el que opera UPnP.

- ◆ Las *Stateful Packet Inspection* (SPI) extensibles y el diseño modular.

Casi todas las funcionalidades del Núcleo Cling pueden ser sobrescritas o remplazadas con código propio. Además los subsistemas de la red de transporte, mensajes *parseado* e incluso cualquier aspecto individual de la implementación del protocolo UPnP ofrece SPI para su extensión y modificación.

- ◆ Soporte Completo para Android

Todas las características del Núcleo Cling son soportadas en Android, la versión mínima es la versión 2.1.

### 3.2. LonWorks

LonWorks [9] es un estándar que fue desarrollado para resolver el problema del control tradicional de las redes. Este estándar es capaz de utilizar gran variedad de medios de transmisión pudiéndose transmitir por el aire, cables de par trenzado, cable coaxial, fibra óptica o la red eléctrica y usa el protocolo de comunicación LonTalk.

Las redes LonWorks se controlan por medio de nodos a lo largo de la red que gestionan los distintos sensores y actuadores. Todos los dispositivos en una red LonWorks suelen implementarse con chip de un Neuron de Echelon, aunque puede usarse cualquier microcontrolador.

El Neuron Chip utiliza un transceptor para enviar y recibir información, sus principales características [10] son las siguientes:

- ◆ El Neuron Chip contiene un identificador univoco denominado *Neuron ID*
- ◆ Los microcontroladores *Neuron Chip* son programados en *Neuron C*, un lenguaje estructurado basado en el estándar ANSI de C.
- ◆ Es independiente del medio de transmisión.

#### 3.2.1. LonTalk

Es un protocolo optimizado para el control que fue desarrollado por la empresa *Echelon Corporation* para dispositivos en red.

Se puede decir que es la cabeza de la Tecnología LonWorks, proporcionando una serie de herramientas de servicios de comunicación que permiten a un dispositivo enviar y recibir mensajes desde otros dispositivos sobre la red sin necesidad de conocer la topología de la Red.

El protocolo está dividido en capas basado en paquetes, siendo un protocolo de comunicaciones punto a punto, como Ethernet y protocolos de Internet. Es un estándar publicado por la ISO/IEC JTC 1/SC 6 y definido en la ISO ISO/IEC 14908.1.

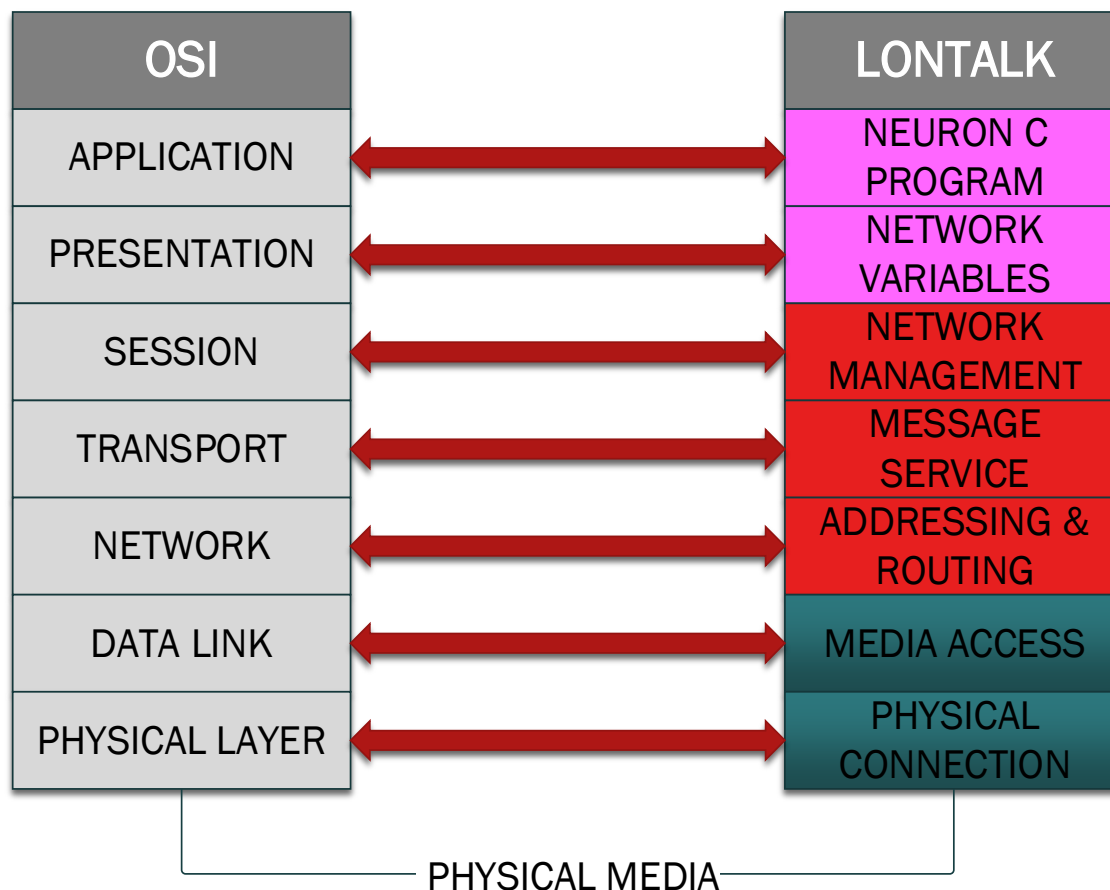


Figura 4 : Lontalk Architecture

### 3.2.2. iLON100

Este producto es proporcionado por la Empresa *Echelon Corporation*, que actualmente se encuentra en la versión 2.0 denominada *SmartServerController* [11].

El dispositivo *iLON100* es un dispositivo que ofrece una interfaz de servicios Web y da acceso desde una red Local, una red privada virtual o desde Internet a los dispositivos LonWorks, que se encuentren conectados. Proporcionando la posibilidad de leer, escribir o configurar sus *data points* [12]

El dispositivo *iLON100* se puede utilizar de *Gateway* para una gran variedad de dispositivos, como por ejemplo los dispositivos *INS-231 F/V3* y *INS-360*, ModBus, M-Bus, y dispositivos I/O digitales.



FUENTE: PFC TECNOLOGÍA APLICADA AL HOGAR DIGITAL

Figura 5 : Dispositivo iLON100 Internet Server

#### 3.2.2.1. Especificaciones funcionales del iLon100 Internet Server

- ◆ Servidor SOAP/XML para la configuración y gestión de aplicaciones internas del nodo.
- ◆ Aplicaciones en lenguajes de Alto Nivel.
- ◆ Posibilidad de personalizar planos 3D en la empresa.
- ◆ Amplias librerías de diseño en 3D.
- ◆ Herramientas de administración intuitivas.

#### 3.2.3. Lonmaker

Lonmaker [13] es una aplicación de diseño y control para el desarrollo, aplicada a la tecnología LonWorks.

Las principales funcionalidades de esta aplicación son las siguientes:

- ◆ Software interoperable y abierto usado en el diseño, instalación y mantenimiento de Redes LonWorks.
- ◆ Soporta Plug-ins LonWorks Network Services (LNS) para configurar los dispositivos de una forma más rápida.
- ◆ Utiliza Visio combinando una arquitectura Cliente - Servidor
- ◆ Incluye la Herramienta Human Machine Interface (HMI) que está basado en componentes integrados para su monitorización.

### 3.3. Android

#### 3.3.1. ¿Qué es Android?

En primer lugar, se define el término plataforma. Una plataforma es un conjunto completo de varias *Application Programming Interface* (API) que constituyen una plataforma software siendo algunas API dependientes del SO.

Android fue desarrollado por Google y es un sistema operativo basado en *Linux*, que posteriormente fue desarrollado por la *Open Handset Alliance*.

La plataforma JAVA es un ejemplo de plataforma no dependiente del S.O. y como JAVA es tanto un lenguaje como una plataforma de desarrollo, esta incluye la JAVA Virtual Machine (JVM) cuya función es interpretar el *bytecode* resultante de la compilación del código fuente de un programa escrito en lenguaje JAVA.

Una parte importante de Android es Dalvik, una máquina virtual basada en la JVM de la plataforma JAVA.

##### 3.3.1.1. Breve Historia

Android nació oficialmente el 5 de Noviembre del año 2007. Aunque, la llegada de los primeros dispositivos al mercado basados en este SO fue en Octubre de 2008.



Sin embargo, no fue hasta una versión preliminar del *Software Development Kit* (SDK) junto con un emulador y alguna documentación para su desarrollo cuando se produjo su crecimiento en el mercado.

### 3.3.2. Principales Características

Las principales características [14]- [15] con las que cuenta el SO Android son las siguientes:

- ◆ **Diseño del dispositivo**

Es capaz de adaptar la resolución y densidad de la pantalla, consta de una gran variedad de bibliotecas de gráficos para 2D y 3D basada en las especificaciones de la OPENGL ES 2.0.
- ◆ **Almacenamiento**

Para almacenar información sobre la Base de Datos se hace uso de la librería *SQLite*, que es una base de datos liviana.
- ◆ **Conectividad**

La conectividad que soporta en la actualidad, está basada en las siguientes tecnologías GSM/EDGE, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+, WiMAX.
- ◆ **Mensajería**

SMS y MMS son formas de mensajería y ahora además se incluye el C2DM que es parte del servicio *Messaging* de Android.
- ◆ **Soporte de JAVA**

Android no cuenta con una JVM como tal en su lugar el *bytecode* de Java corre en la Máquina Virtual Dalvik, la cual es una máquina virtual diseñada específicamente para Android.

Por otro lado, el soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME *MIDP Runner*.

◆ *Streaming*

Android tiene soporte para *Streaming* RTP/RTSP (3GPP PSS, ISMA), HTML (HTML 5).

◆ Entorno de desarrollo

El entorno de desarrollo incluye un emulador de dispositivos, herramientas para su depuración que posibilitan el análisis del rendimiento software.

El entorno de desarrollo usado en el proyecto es Eclipse [16], que ofrece un plugin gratuito para integrar el SDK de Android.

### 3.3.3. Evolución



Figura 6 : Evolución del Sistema Android

Como se observa en la Figura 6 la evolución del SO Android ha sido constante. Desde su primera versión, allá por el 2008, el sistema se ha ido revisando contando con nuevas funcionalidades a lo largo de este periodo.

Con la llegada de cada nueva versión del SO, se le da un nuevo nombre a esa nueva versión, aludiendo el nombre a una comida dulce o postre, siendo además su primera letra acorde con el orden alfabético.

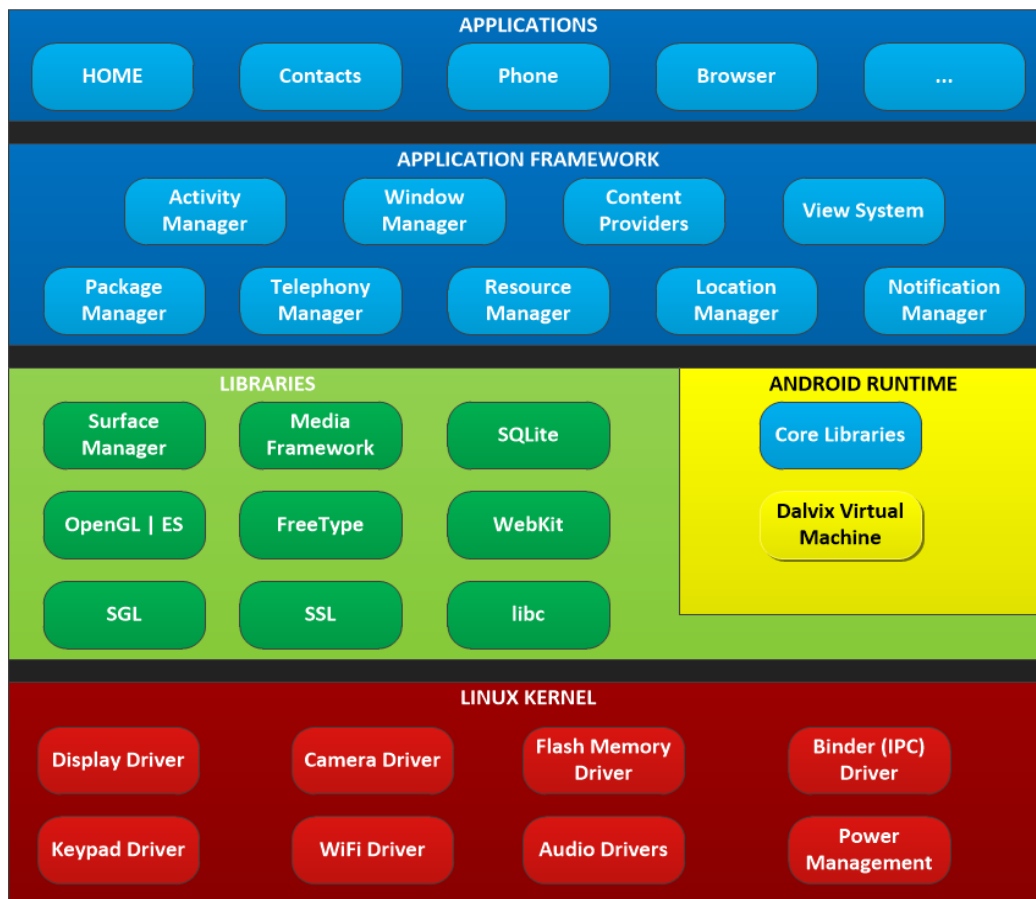
En la versión 1.5 se mejoraron las transiciones de ventanas y el *scroll*. Con la llegada de la versión 2.0 llegó la expansión a nivel mundial de este SO con un cambio casi completo en todos sus componentes visuales, iconos, menús, etc. Trayendo consigo los famosos fondos animados de pantalla que fue una gran novedad en su lanzamiento.

Más adelante, con la llegada de la versión 3.0 se realizó un nuevo rediseño de la interfaz no siendo tan implantada en los nuevos dispositivos móviles, debido a que esta versión fue utilizada como puente para la optimización de pantallas con resoluciones de 1280x800 píxeles en Tablets y Smartphones.

La última versión hasta la fecha, la versión 4.0 ha traído consigo la unión de todas las ramas del sistema, sufriendo nuevamente un rediseño de la interfaz del SO y que las aplicaciones puedan ser divididas en módulos denominados fragmentos consiguiendo que una aplicación pueda contener varios fragmentos.

### 3.3.4. Arquitectura

La Arquitectura de Android está basada en un sistema por capas. Utiliza el *Kernel de Linux 2.6*, que le brinda el acceso a la parte *hardware*, a la par que le permite ser compatible con la mayoría de los drivers.



FUENTE: [HTTP://DEVELOPER.ANDROID.COM/ABOUT/VERSIONS/INDEX.HTML](http://developer.android.com/about/versions/index.html)

Figura 7 : Arquitectura Android

- ◆ **Capa de Aplicaciones**  
 Las aplicaciones pueden estar compuestas por cinco bloques: *Activity*, *Intent*, *Broadcast*, *Services* y *Content Providers*, no siendo necesarias todas en una misma aplicación.  
 En las aplicaciones por defecto del sistema se pueden encontrar clientes de correo, navegador web, mapas, calendarios, etc.
- ◆ **Framework**  
 La arquitectura mostrada en la Figura 7 está diseñada para que la reutilización de componentes sea sencilla. Esto se consigue gracias a que es necesario el cumplimiento de normas de seguridad impuestas por el Framework, consiguiendo así alcanzar un importante grado de reutilización de código.

#### ◆ Bibliotecas

Existe un conjunto de librerías en C y C++ que proporcionan la mayor parte de las funcionalidades presentes que son utilizadas por varios de los componentes del sistema Android. Ofrecido a los desarrolladores por medio del Framework.

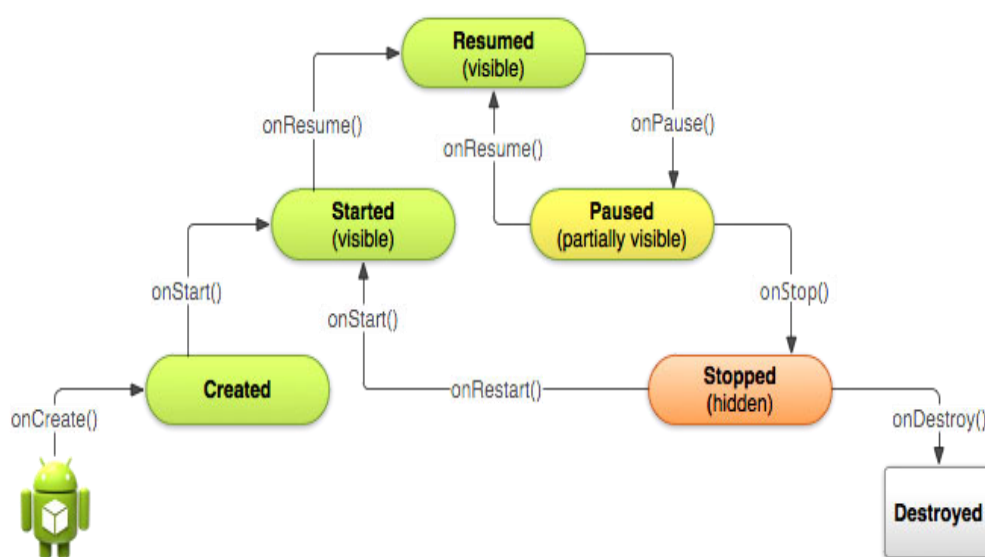
#### ◆ *Android Runtime*

El entorno de ejecución está compuesto por librerías que proporcionan la mayor parte de las funcionalidades de Java incluyendo la JVM. Al ejecutarse, cada proceso invoca y crea su propio proceso con su propia instancia de la máquina virtual Dalvik.

La Máquina virtual Dalvik ha sido diseñada de manera que un mismo dispositivo pueda lanzar múltiples instancias de forma eficiente.

### 3.3.5. Ciclo de vida de una Aplicación

Una característica fundamental de Android es que el ciclo de una aplicación no está controlado por la propia aplicación, sino que lo determina el sistema a partir de una combinación de estados. La combinación de estados del ciclo de vida de una aplicación se puede ver con más detalle en la Figura 8.



FUENTE: [HTTP://DEVELOPER.ANDROID.COM/TRAINING/BASICS/ACTIVITY-LIFECYCLE/STARTING.HTML](http://developer.android.com/training/basics/activity-lifecycle/starting.html)

Figura 8 : Ciclo de Vida de una Aplicación

## 4. Antecedentes y Análisis del Sistema

---

Se desea desarrollar un sistema que nos permita integrar los elementos de una Red LonWorks como dispositivos en una Red UPnP, integrándose todo ello como una aplicación basada en el SO Android. La aplicación que se ha construido está hecha para la versión 2.3 Gingerbread de Android en la cual se basa todo el diseño y pruebas del Sistema, aunque se ha probado también en las últimas versiones del SO.

### 4.1. Experiencias Relacionadas

Al comienzo de este proyecto se planteó dar solución a la integración de las tecnologías UPnP y LonWorks a través del uso del Framework OSGI apoyándose en la experiencia descrita en el documento “Integrate LonWorks and UPnP devices by OSGI gateway and universal remote console” [17]. En este documento sus autores explican cómo integrar las tecnologías UPnP y LonWorks por medio del uso del Framework OSGI. Existen varias implementaciones del *framework* OSGI [18] . Una de esas implementaciones es el contenedor *Knopflerfish* que cuenta con una serie de tutoriales para ser instalado en el S.O. Android, siendo el inicialmente elegido.

Sin embargo, se encontró un problema en la puesta en marcha de este Framework que imposibilitó el uso de OSGI.

Después de consultar con el personal de soporte de la empresa Echelon se supo que OSGI ya no era soportado. Por tanto, no se ha podido utilizar el Framework OSGI para resolver la problemática de la integración de las tecnologías UPnP y LonWorks.

## 4.2. Análisis de especificaciones

Para conseguir los objetivos de este proyecto expuestos en el apartado 1.2 es necesario que se alcanzasen los siguientes requisitos, siendo prioritario solucionar la problemática de la integración de las tecnologías UPnP y LonWorks.

### 4.2.1. Requisitos funcionales

Se hará uso de diagramas UML para realizar un análisis de los requisitos funcionales. Gracias a estos diagramas se puede ver el funcionamiento del sistema previsto en situaciones específicas. Además se ilustra la aplicación y su uso en el Anexo D.

El lenguaje de modelado utilizado es UML, el cual es el lenguaje de facto en la industria actualmente. Es un lenguaje gráfico que permite visualizar, especificar, construir y documentar un Sistema.

#### 4.2.1.1. Diagramas de Casos de Uso

La interacción que tiene un usuario con el sistema es representado por medio de los diagrama de casos de uso.

En los siguientes diagramas de casos de uso se representa los requisitos del sistema.

La Figura 9 muestra como el usuario pone en marcha el sistema tras un primer uso inicial y con esto se podrá controlar los dispositivos del área que usuario desee

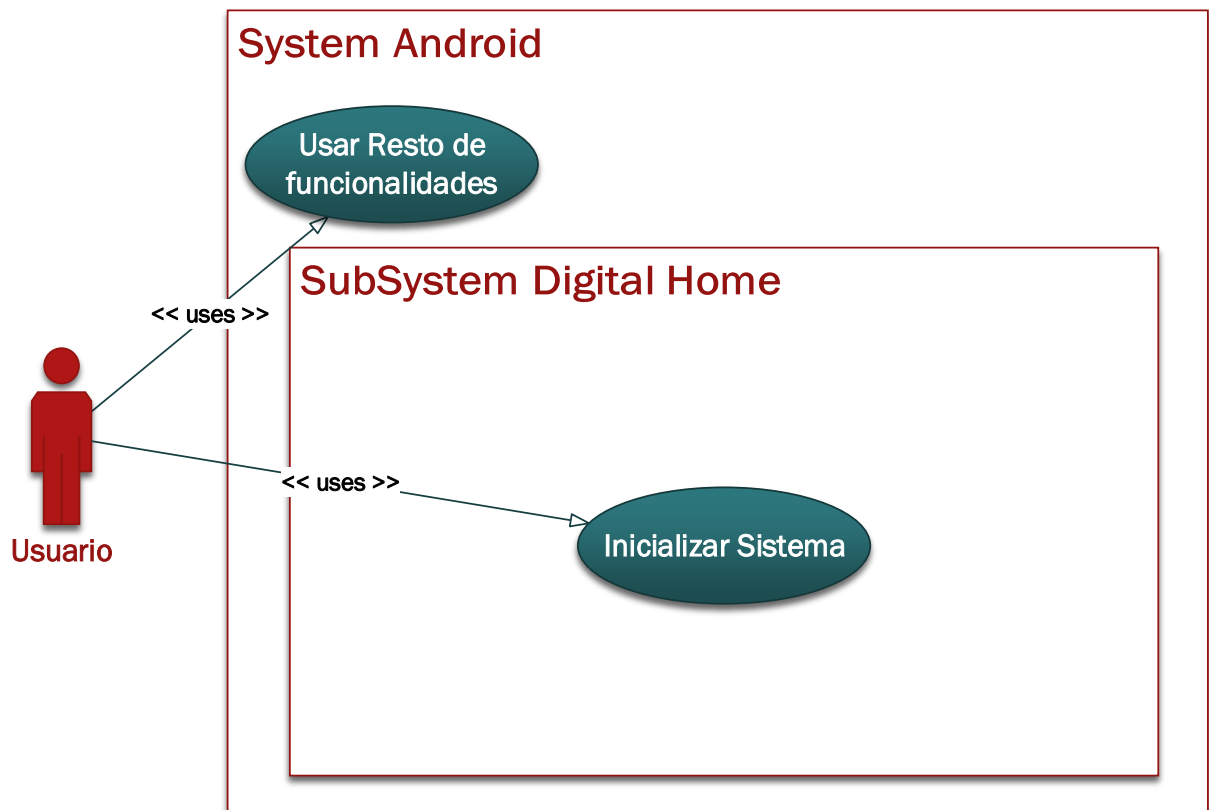


Figura 9 : Diagrama de Casos de Uso Principal

En la Figura 10 se observa que dentro del caso de uso *Inicializar Sistema* se definen seis casos de uso correspondientes al control de dispositivos UPnP, cada uno de ellos pertenece a una zona o área.

En las áreas Entrada, Dormitorio, Baño, Cocina y Salón aparecerán los dispositivos que se puedan controlar y clasificar según los campos de información encontrados en la búsqueda de dispositivos UPnP en la Red LonWorks y/o los ficheros de configuración. La área Otros contendrá todos los dispositivos restantes incluyendo los dispositivos UPnP remotos que son reconocidos gracias a la UPnP Stack.



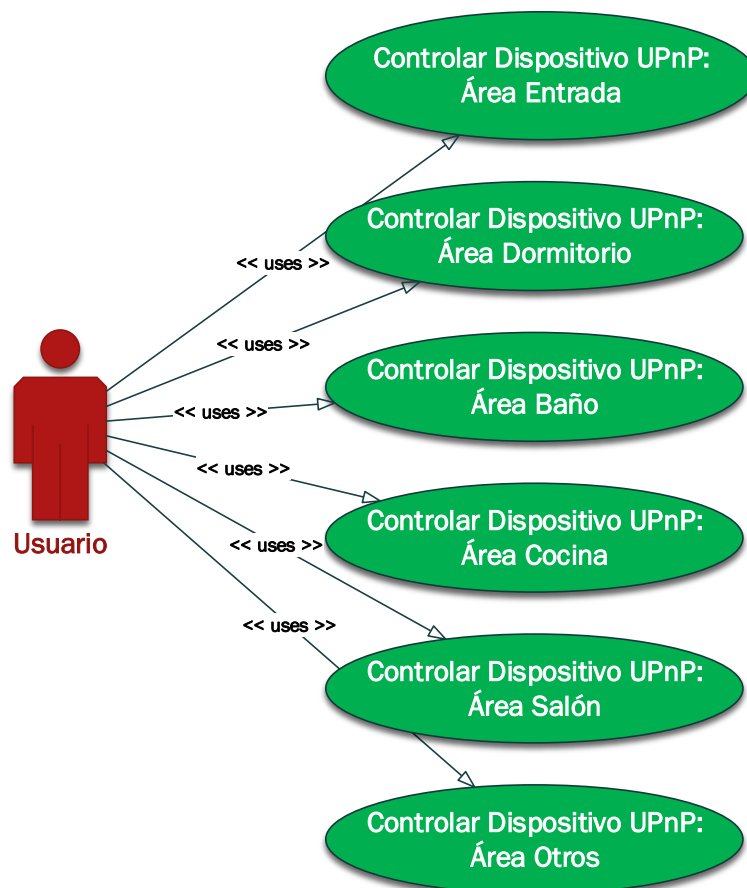


Figura 10: Diagrama de casos de uso Inicializar Sistema

Los casos de uso de la Figura 10 contienen un caso de uso interno que es el de Invocar Acciones y se ilustra en la Figura 11. Esta figura representa el diagrama de casos de uso Controlar Dispositivos UPnP para el área de la entrada, aunque todas las áreas contienen el mismo caso de uso Invocar Acciones.

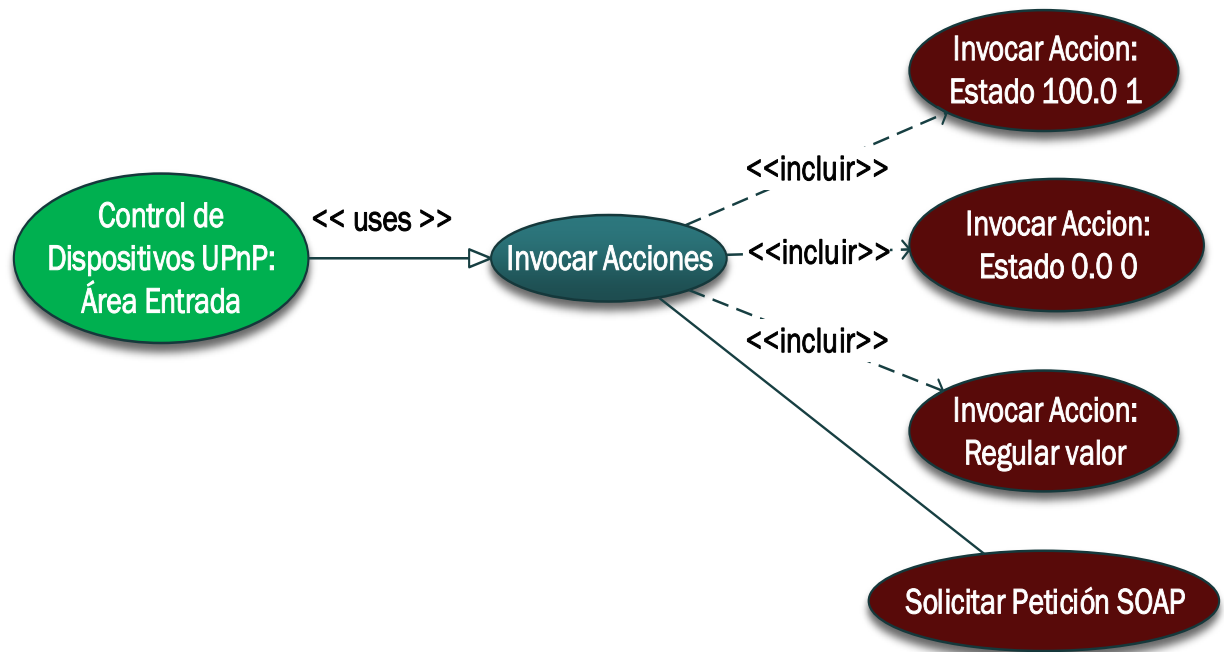


Figura 11: Diagramas de Casos de Uso - Control de Dispositivo UPnP



## 5. Diseño del Sistema

---

Como ya se comentó se va a construir una aplicación Android en la versión 2.3 (Gingerbread). La gran mayoría de aplicaciones Android utiliza el patrón de arquitectura llamado Modelo Vista Controlador (MVC), la principal ventaja de este patrón es que separa los datos de una aplicación, la lógica de negocios y la interfaz de usuario. Se ha utilizado este patrón para el desarrollo de la aplicación Android.

Las funcionalidades que tiene que cumplir el Sistema son las siguientes:

- ◆ Iniciar los servicios necesarios que proporcionen al sistema la *UPnP Stack*.
- ◆ El sistema debe ser capaz de crear una Red UPnP con los servicios proporcionados con la *UPnP Stack*.
- ◆ Debe de ser capaz de reconocer y visualizar los dispositivos compatibles UPnP de la Red LonWorks.
- ◆ Interaccionar con el *iLon100 Internet Server* usando el intercambio de mensajes SOAP.
- ◆ Tiene que ser capaz de visualizar y controlar los elementos de la Red LonWorks tratándolos como Dispositivos UPnP.

### 5.1. Entorno

En la Figura 12 se ve el escenario físico del sistema, en la figura se ve un dispositivo móvil Android versión Gingerbread que se conecta al Router Wireless a través de una conexión WiFi en un entorno Wide Area Network. Con esto se consigue que el dispositivo Android pueda interactuar con el dispositivo iLon100 Smart Server y con ello con los dispositivos LonWorks del entorno domótico proporcionado por los Paneles Demostrativos.



Figura 12 : Escenario Físico

Con la Figura 13 se representa la arquitectura del sistema utilizando las tecnologías UPnP y LonWorks y todo ello unido y visto a través del dispositivo Android.

La Red LonWorks tiene los dispositivos INS-231 y INS-360 que contienen los elementos del Hogar que son configurados a través de su data *points* como las luces, persiana, detectores, sensores, etc.

Los elementos del Hogar son configurados modificando sus *data points* gracias a la ayuda del dispositivo iLON100 y su interfaz SOAP/XML a través del intercambio de mensajes SOAP se construye la Red UPnP transformando los elementos del Hogar en dispositivos UPnP.

La Red UPnP se integra en la Aplicación Android gracias a la ayuda de los servicios de la UPnP Stack proporcionados por las librerías y herramientas Cling.

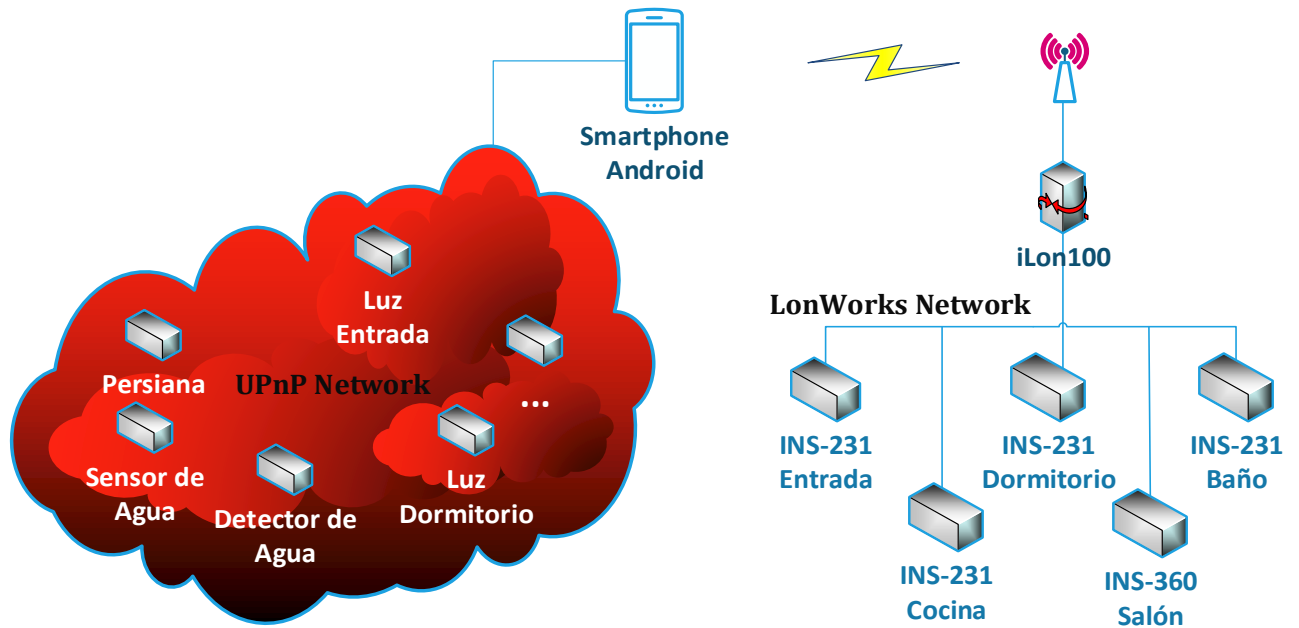


Figura 13: Arquitectura Sistema

## 5.2. Servicios UPnP

Los Servicios UPnP son utilizados para crear la Red UPnP, añadir dispositivos y puntos control, todo ello se da gracias al uso de la *UPnP Stack*. Para contar con la UPnP Stack se hace uso de las librerías Cling que implementa la Arquitectura UPnP. La librería Cling se encuentra dentro de los proyectos de software libre desarrollados por *4th Line*.

La aplicación cargará el servicio UPnP para crear una Red UPnP agregando los dispositivos compatibles UPnP que encuentre en la red a la que esté conectada el Smartphone.

En este proyecto no se contempla el control de los dispositivos remotos reconocidos. Es decir, que cuando se cargue el servicio UPnP se reconocerán no solo los dispositivos UPnP creados a partir de los elementos LonWorks sino también los dispositivos reconocidos que cumplan con la descripción de dispositivo UPnP denominados dispositivos remotos. Aunque, se puede ver en el apartado 0 que esto se contempla como una posible línea futura a seguir.

### 5.3. Configuración de la Red LonWorks con la herramienta de diseño Lonmaker.

Para implementar la Red LonWorks es necesario configurar en primer lugar los parámetros que tendrá la red que se va a utilizar. Todos los parámetros y pasos a seguir se indican en detalle en la guía de usuario de Lonmaker [13].

Para configurar la red LonWorks se selecciona en el menú de arranque *Network Interface* dándosele un nombre específico a la red. A continuación se selecciona la opción *OnNet* para que los cambios que se realizan, se registren durante su compilación.

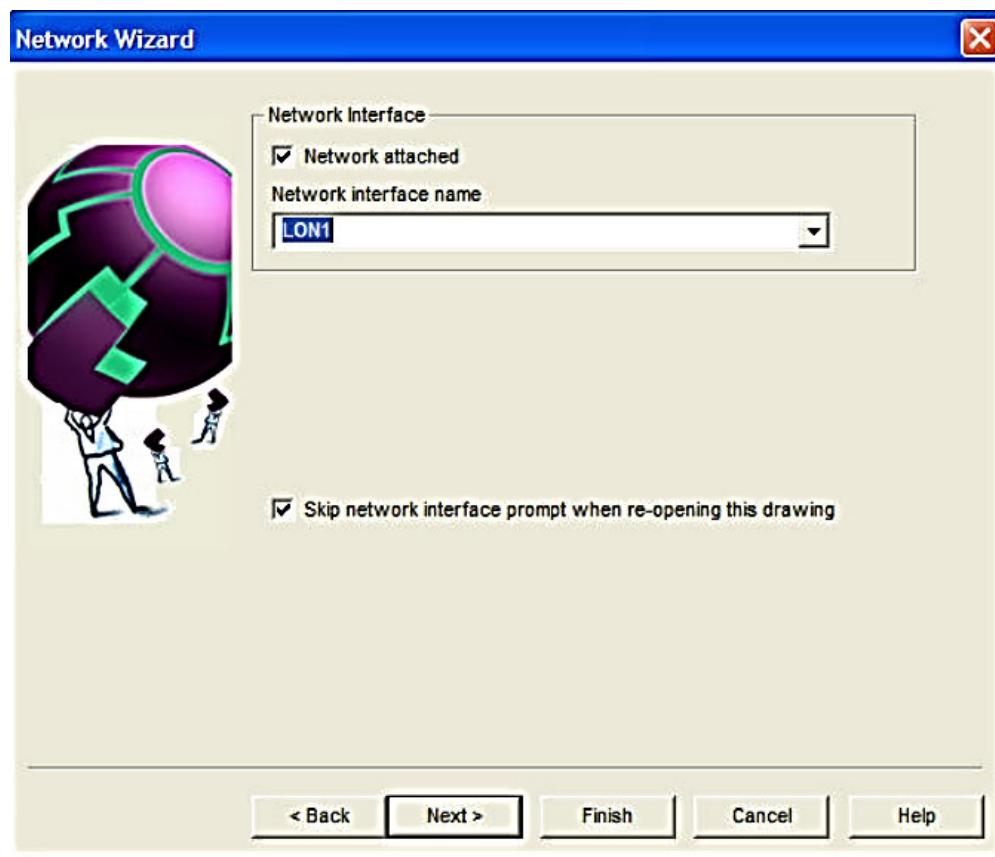


Figura 14 : Lonmaker Network Design

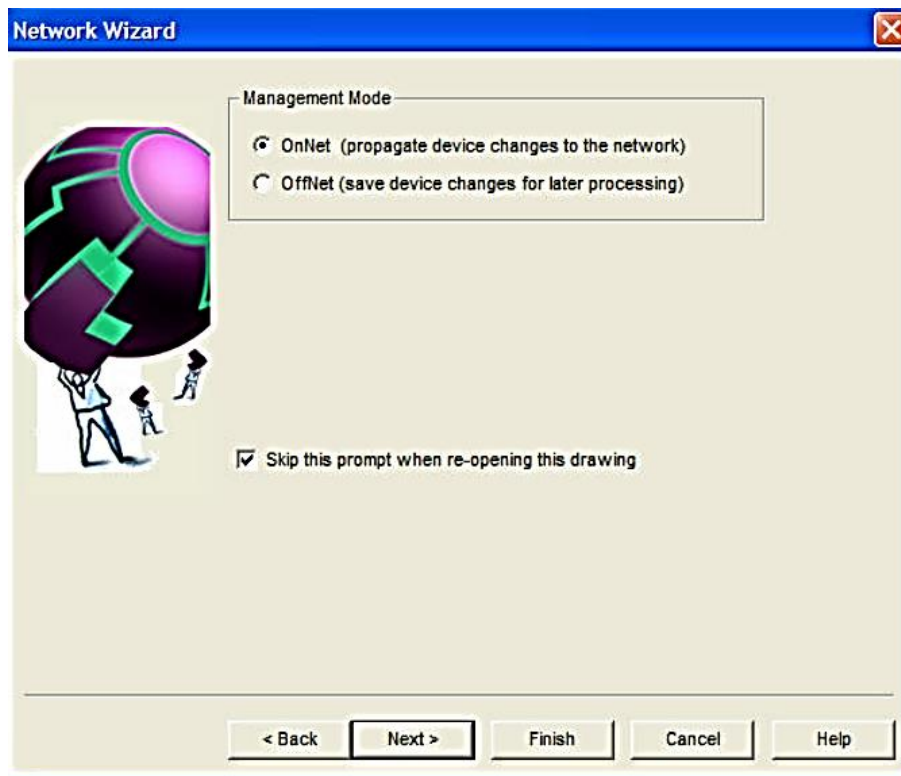


Figura 15 : Lonmaker Network Design - 2

A continuación, se añade los dispositivos seleccionando en *New Device Wizard* y se comisiona los dispositivos (darlo de alta). Después se crea un bloque funcional con las variables de red por defecto que ofrezca el dispositivo.

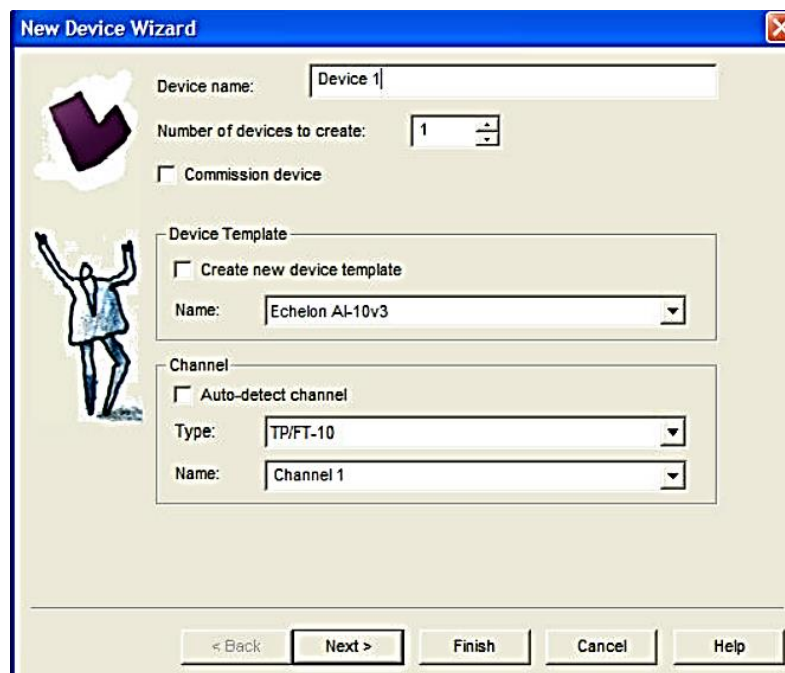


Figura 16 : New Device Wizard



Para implementar la Red LonWorks es necesario conectar las variables de red de salida “*nvo*” y de entrada “*nvi*” [19] pertenecientes a Local data points (NVL) [20] y con las cuales se lee, escribe y configura los elementos LonWorks del Hogar.

La conexión de las variables depende de cómo se quiera que se comporte la Red LonWorks. En este PFC se ha realizado configuración que se puede ver en la Figura 17 y en Figura 18.

En primer lugar en la Figura 17 se puede ver como por medio del dispositivo *iLon100* descrito en el apartado 3.2.2 se ha hecho uso del bloque funcional llamado *ILON.SNVT\_switch*, conectando sus variables de red de entrada y de salida con los bloques funcionales correspondientes a los dispositivos *INS-231* y *INS-360*.

Con esto se consigue que al modificar el valor de las variables de red del bloque funcional *ILON.SNVT\_switch* se modifique el valor de las variables de los dispositivos *INS-231* o *INS-360* que se encuentren conectadas a las variables de red del *ILON.SNVT\_switch* a las que se puede acceder a través del Nodo Servidor *iLON100*.

A modo ilustrativo se expone la Figura 18 para ver los dispositivos LonWorks que se encuentran en los Paneles Demostrativos y como sería la configuración en las Áreas del Salón y la Cocina del Hogar

Como se verá más adelante en la Figura 19 su contenido será dependiente de cómo se configure las conexiones entre los dispositivos *INS-231* e *INS-360* correspondientes a las áreas de los *Paneles Demostrativos*. Las conexiones que se han establecido para este Sistema se analizan con más detalle en el Anexo C.

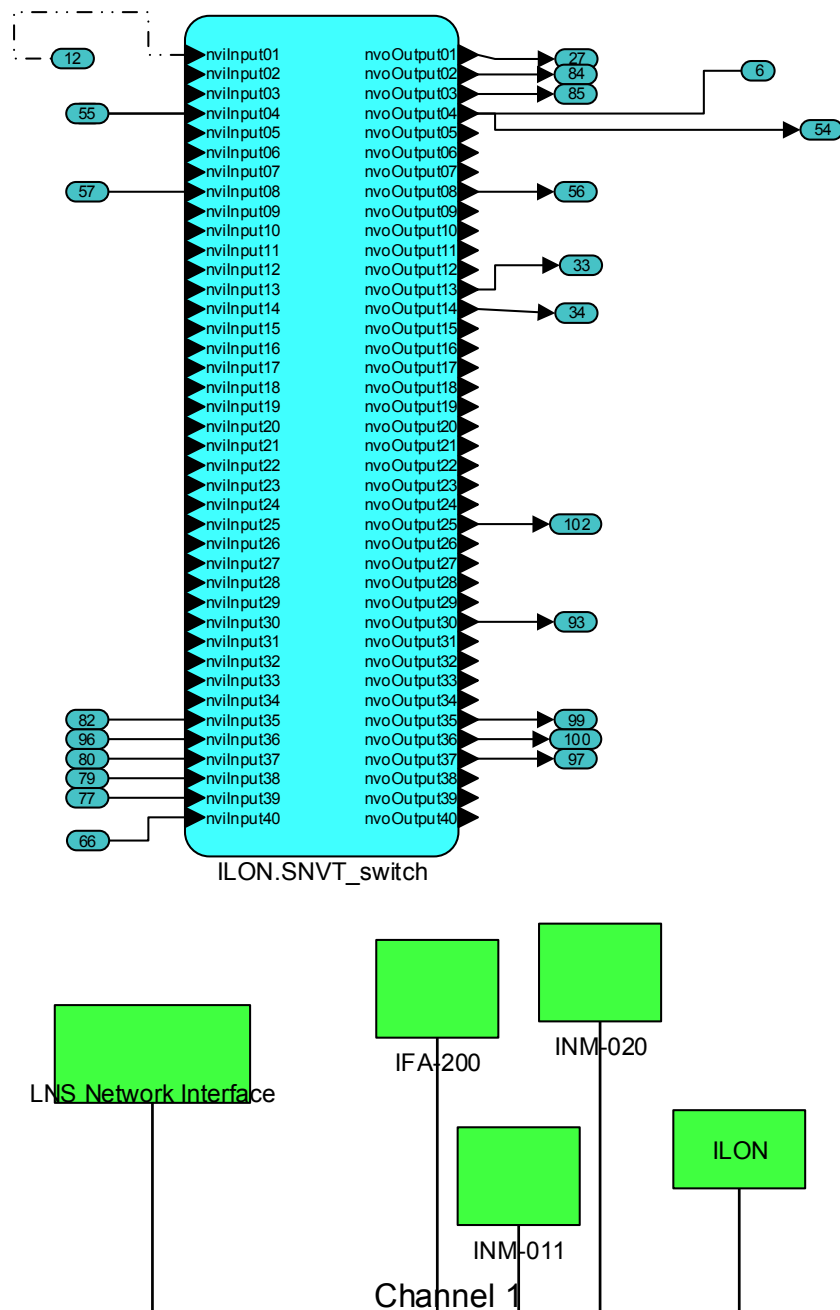
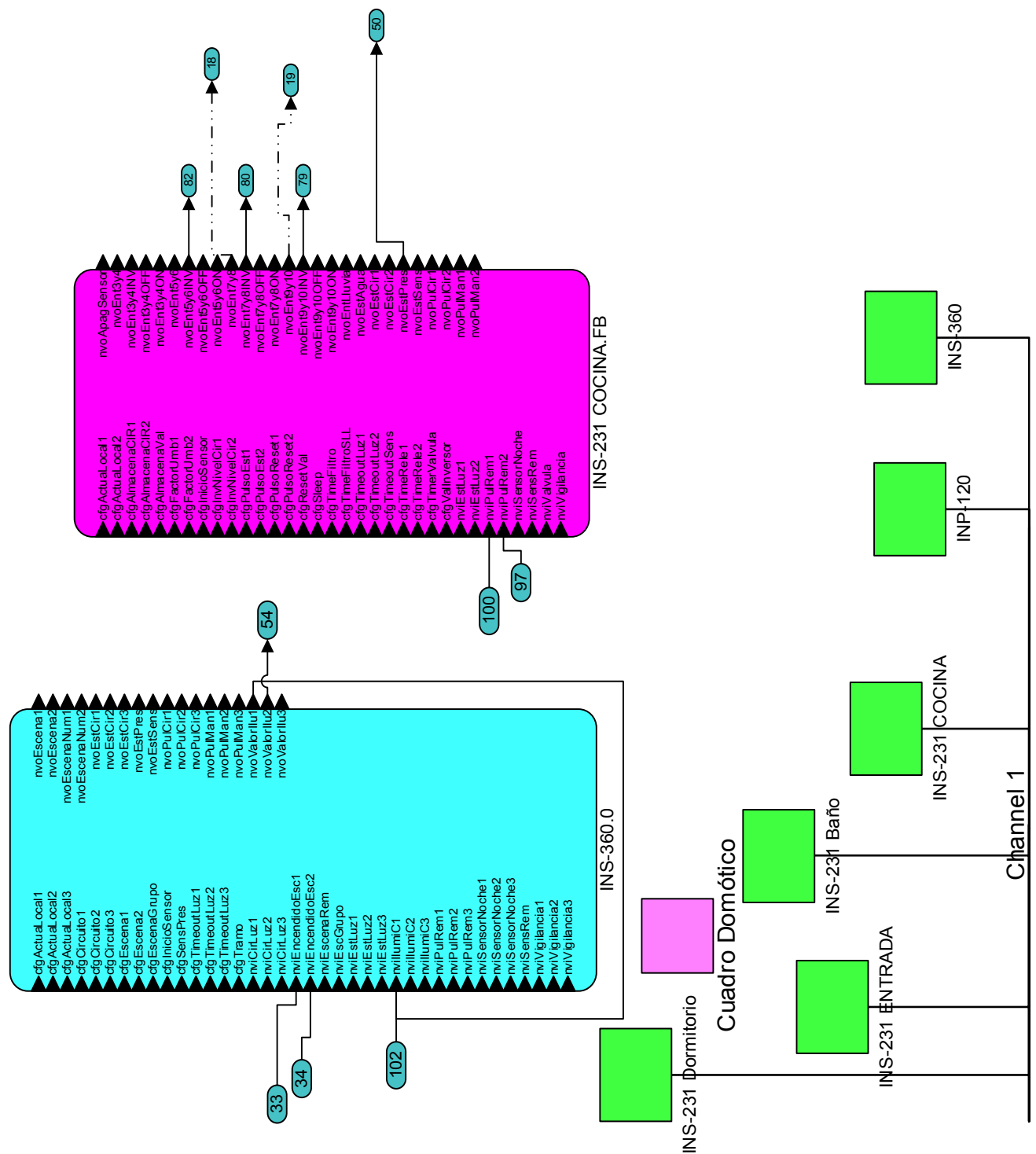


Figura 17 : Configuración Cuadro Domótico



**Figura 18 : Configuración Dispositivos de las Áreas (Entrada, Salón, Baño, Cocina y Dormitorio).**

## 5.4. Integración de elementos LonWorks como Dispositivos UPnP

La solución a la problemática de integrar las tecnologías UPnP y LonWorks fue intercambiar mensajes SOAP y con el análisis de las respuestas de dicho intercambio poder conocer los elementos del hogar. Es decir, conocer los elementos de los dispositivos LonWorks y los estados de sus variables. Los mensajes que se utilizan se proporcionan a través de la interfaz SOAP, como por ejemplo el tipo de mensaje *"DataServerWrite"* [21].

Además, se cuenta con un fichero WSDL que define dicha interfaz SOAP/XML y se hace uso de la versión 1.1 en este Proyecto. El contenido del fichero WSDL se puede ver en el Anexo A apartado 8.1.

Los dispositivos Locales reconocidos se verán igual que si fuera un dispositivo remoto pero los dispositivos locales serán creados a partir de los elementos de cada dispositivo LonWorks como dispositivos UPnP. Estos elementos LonWorks se filtrarán y se gestionarán con el uso de ficheros de configuración creados a partir de la configuración realizada por medio de la herramienta Lonmaker tal como se ha descrito en el apartado 5.3. Los ficheros usados tienen formato XML.

Los ficheros de configuración indican al sistema qué elementos pueden ser tratados como dispositivos UPnP y partir de eso se creara los componentes necesarios para el control del estado de las variables de los dispositivos creados en la aplicación Android.

Por otro lado, La creación de los Dispositivos UPnP se solucionó gracias a la ayuda de la herramienta Cling estudiada en el apartado 0, permitiéndonos así tener un sistema con las funcionalidades de la UPnP Stack y con esto crear dispositivos UPnP únicamente con la condición de formarlos correctamente según el *Documento de Descripción UPnP* que se puede ver en el apartado 3.1.1.1.

Ningún hogar es igual a otro siendo necesario que el sistema a implementar sea capaz de adaptarse al entorno en el que se encuentre. Por ello se han creado una serie de interfaces que se explican con más en detalle en el Anexo B, apartado 9.2 con el

objetivo de conseguir que el sistema se adapte a cualquier entorno domótico que use la tecnología LonWorks, permitiendo crear correctamente los ficheros de configuración tras el despliegue de la Red LonWorks.

El uso de los ficheros de configuración en este PFC se ilustra en la Figura 19 con la creación del fichero "*home\_config.xml*".

El fichero "*home\_config.xml*" se encuentra en el paquete "*res/raw/*". El paquete *res* es un paquete que tienen todas las aplicaciones Android y es dónde se almacenan los recursos utilizados por la aplicación. En el documento *Web Resources Types* [22] se describen con detalle el formato de los recursos de una aplicación Android.

En la Figura 19 se describe el contenido del fichero *home\_config.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<devices>
  <!-- LUCES -->
  <device><!-- Entrada -->
    <pointname>NVL_nvoOutput01_199</pointname><!-- elemento obligatorio -->
    <icon>2130837508</icon>          <!-- elemento obligatorio -->
    <area>hall</area>                <!-- elemento opcional -->
  </device>
  <device><!-- Dormitorio -->
    <pointname>NVL_nvoOutput02_199</pointname>
    <icon>2130837508</icon>
    <area>bedroom</area>
  </device>
  <device><!-- BAÑO!!! -->
    <pointname>NVL_nvoOutput03_199</pointname>
    <icon>2130837508</icon>
    <area>bathroom</area>
  </device>
  <device><!-- SALON -->
    <pointname>NVL_nvoOutput25_199</pointname>
    <icon>2130837508</icon>
    <area>livingroom</area>
  </device>
  <!-- SENSORES -->
  <device><!-- Sensor de Presencia Entrada -->
    <pointname>NVL_nviInput40_199</pointname>
    <icon>2130837513</icon>
    <sensor>on</sensor>
    <area>hall</area>
  </device>
```

Figura 19 : Formato fichero *home\_config.xml*

En la Figura 19 se observa que para declarar un elemento LonWorks como dispositivo UPnP es necesario declarar los campos "*pointname*" e "*icon*". El campo "*icon*" puede estar repetido con todas las imágenes que pueda contener el dispositivo a crear.

La única condición impuesta por el sistema es que el elemento LonWorks declarado se encuentre en el diseño de la configuración de la Red LonWorks realizada por medio de la herramienta de diseño Lonmaker.

El fichero también incluye algunos campos opcionales que sirven para formar el dispositivo, como es el caso del elemento *"area"* que es declarado por si el elemento LonWorks no tiene el campo *UCPTdescription*. Otro campo opcional es *"index"*.

Para la creación y modificación de este fichero únicamente se necesita conocer, el diseño que tiene la Red LonWorks en el hogar Digital pertinente. Abstrayendo a un posible usuario de tener que conocer cómo se implementa este Sistema.

Por otro lado se hace uso de varias interfaces que pueden ayudar a futuros desarrolladores del Sistema a implementar más servicios y tratar nuevos modelos de datos que se comentara como una posible línea futura a seguir. Estas interfaces se comentaran con más detalle en el Anexo B: Manual del desarrollador.

## 5.5. Control de Dispositivos UPnP

Si el usuario quiere controlar un dispositivo UPnP, se creara un punto de control por medio del servicio UPnP y se buscará ese dispositivo en la Red UPnP con el uso de su identificador unívoco.

Esta funcionalidad se lleva a cabo creando internamente un punto de control que busca el dispositivo seleccionado, y, si el dispositivo se puede controlar<sup>1</sup>, se buscarán los servicios que ofrece el dispositivo y se pondrán en funcionamiento los controladores necesarios en la aplicación Android para poder controlar dicho dispositivo. Esto último se traduce en la implementación de los componentes *Listener* en la aplicación Android.

Por cada dispositivo UPnP se utiliza un componente denominado *ImageView* para que se muestre dicho dispositivo en la aplicación Android. El cual se utilizara para interactuar e invocar acciones de un dispositivo UPnP, generando cambios en los

---

<sup>1</sup> En este PFC no se ha contemplado el control de los dispositivos remotos reconocidos, es decir, solo se pueden controlar los dispositivos locales.

estados de las variables del dispositivo UPnP además de haciendo uso del intercambio de mensajes SOAP para que se actualice los estados del elemento LonWorks en la Red LonWorks. Es decir, cuando se interactúe con un dispositivo UPnP que pertenezca a la Red LonWorks, se actualizará los estados de las variables del dispositivo UPnP y los estados de las variables del elemento LonWorks correspondiente al dispositivo LonWorks al que pertenezca.

## **5.6. Diseño de Componentes Software**

El modelado que se ha realizado está orientado a objetos, permitiendo definir la estructura y el comportamiento que tendrá la aplicación Android. Ayudando a identificar las clases que están implicadas, su comportamiento y la interacción que presentan.

### **5.6.1. Diagramas de Secuencia**

Los diagramas de Secuencia describen la secuencia de sucesos que transcurren en la ejecución concreta de un sistema, este diagrama contiene los detalles de implementación del Sistema.

El diagrama contiene los objetos que intervienen, la interacción que llevan a cabo mostrando el flujo que lleva a cabo el Sistema.

### 5.6.1.1. Cargar UPnP Stack

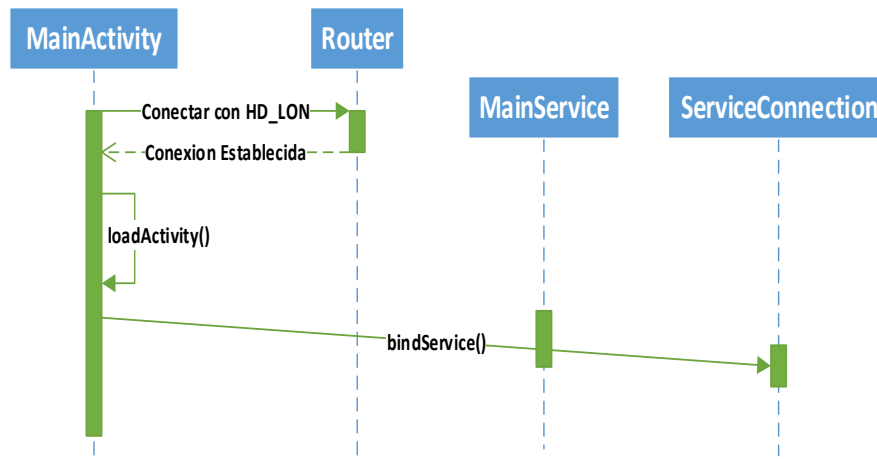


Figura 20 : Diagrama de Secuencia - Cargar Pila UPnP

### 5.6.1.2. Buscar Elementos LonWorks y Crear Dispositivos UPnP

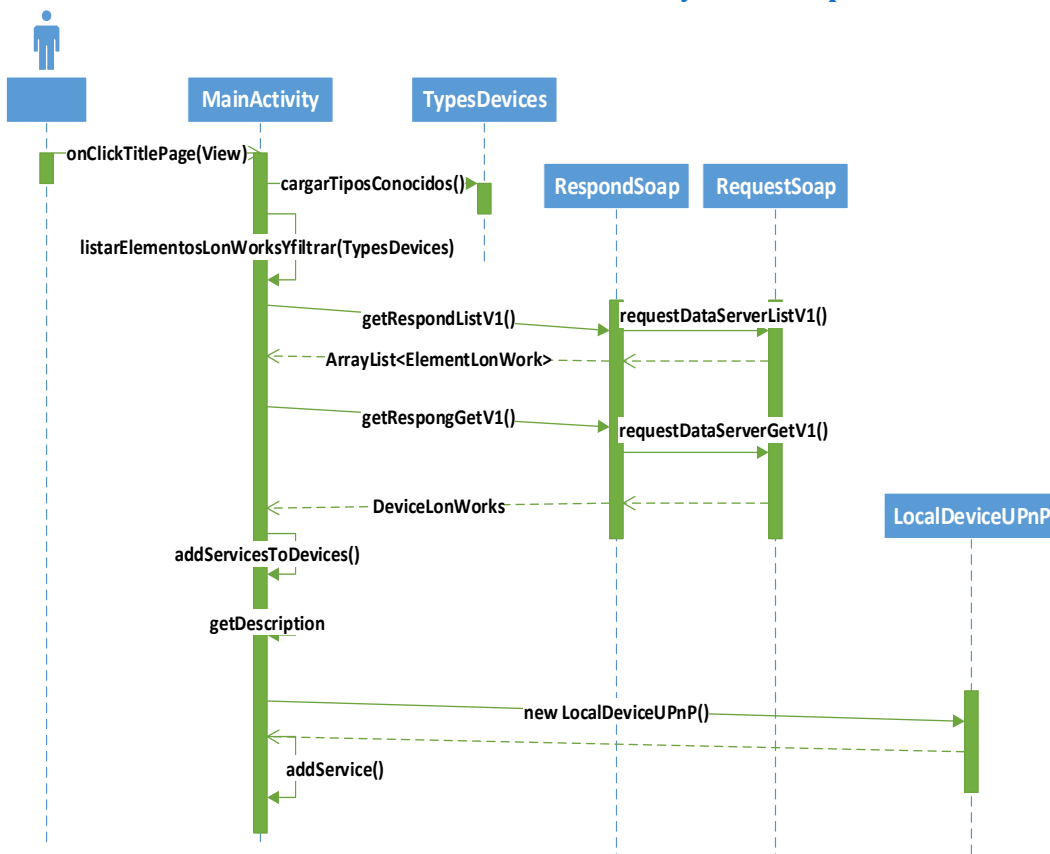


Figura 21 : Diagrama de Secuencia - Dispositivo UPnP desde elemento LonWorks



### 5.6.1.3. Visualizar Área con Dispositivos UPnP

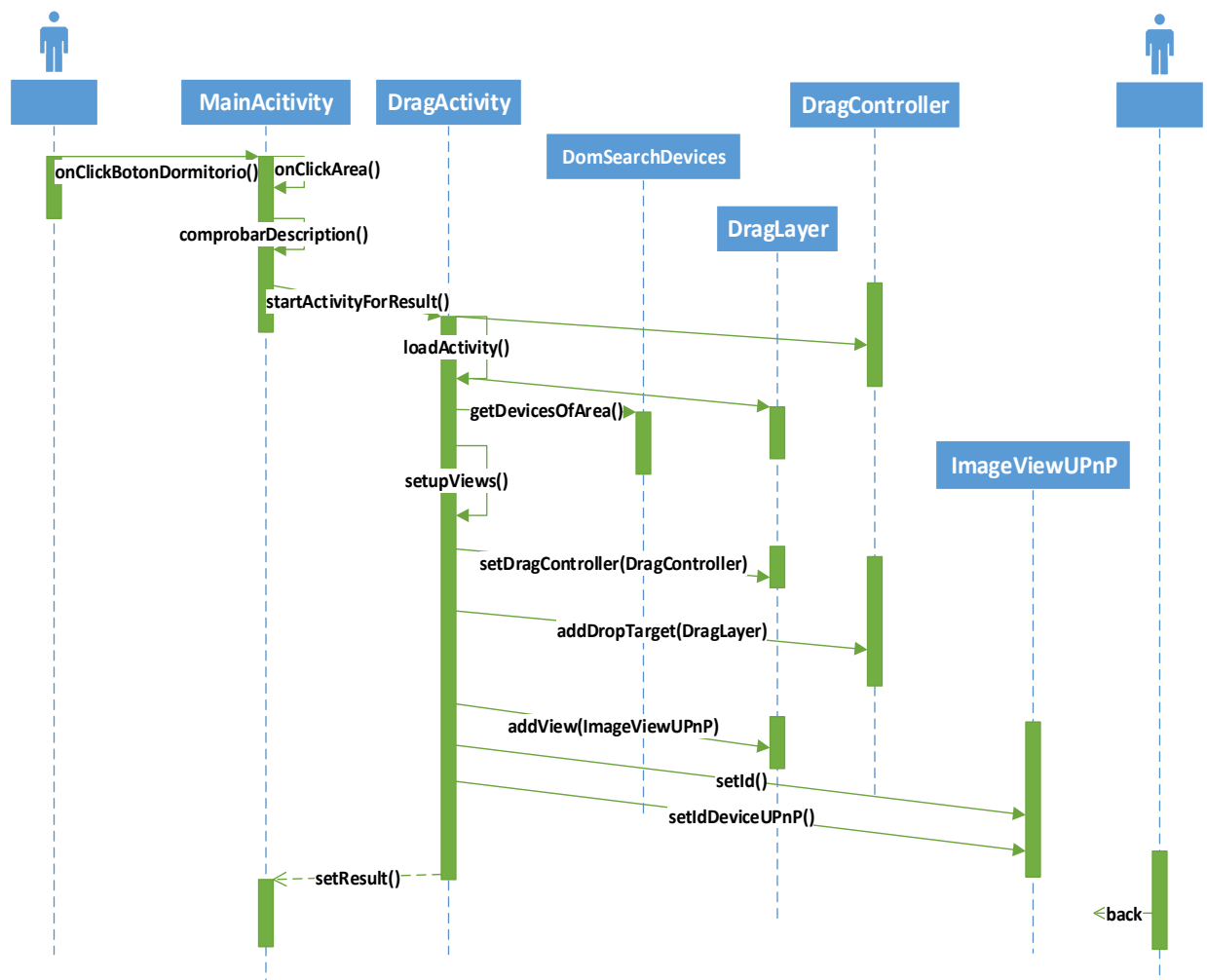


Figura 22 : Diagrama de Secuencia - Área con Dispositivos UPnP

#### 5.6.1.4. Invocar Acciones de un Dispositivo UPnP

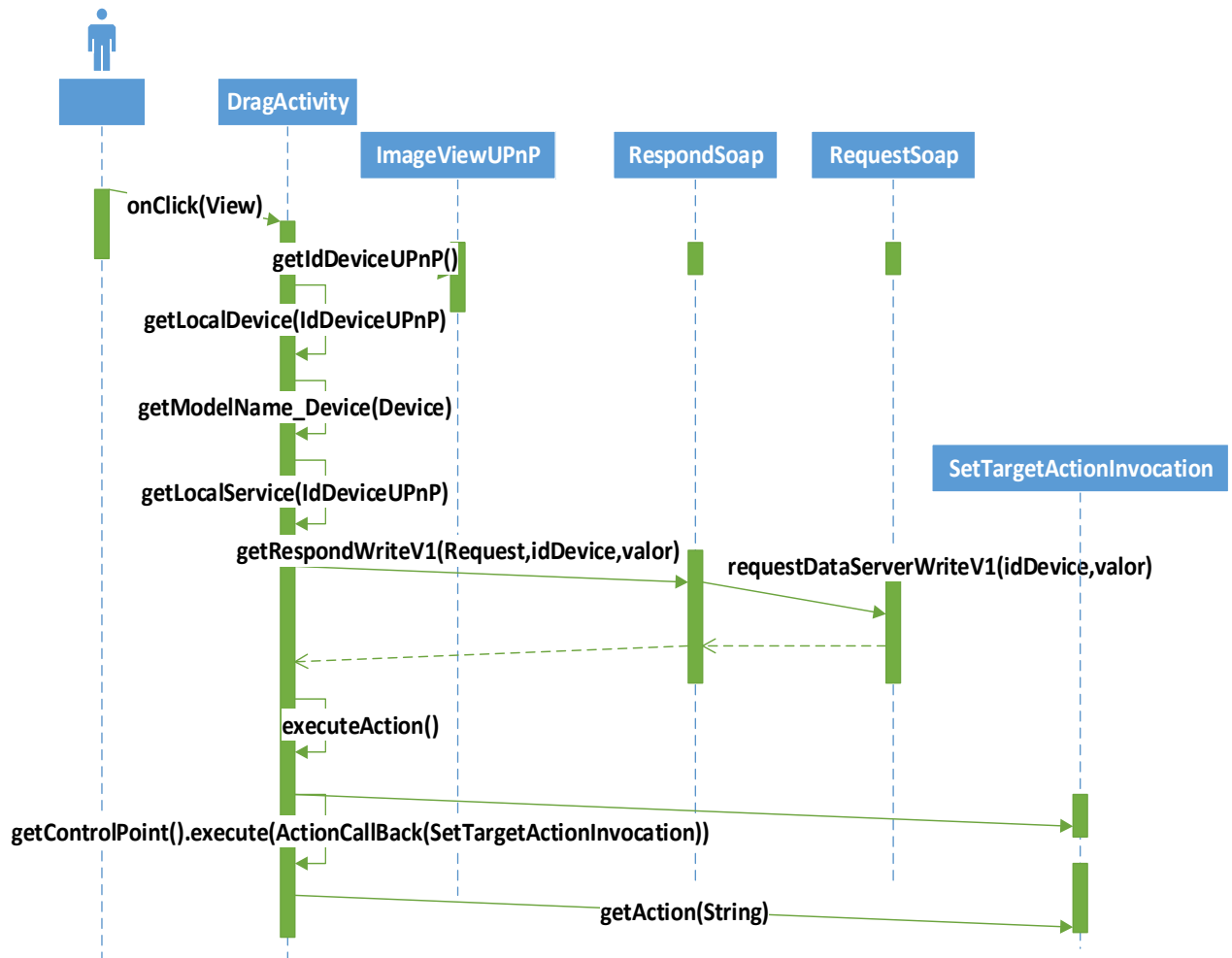


Figura 23 : Diagrama de Secuencia - Invocar Acciones

### 5.6.1.5. Recoger Eventos generados por las Acciones de un dispositivo UPnP

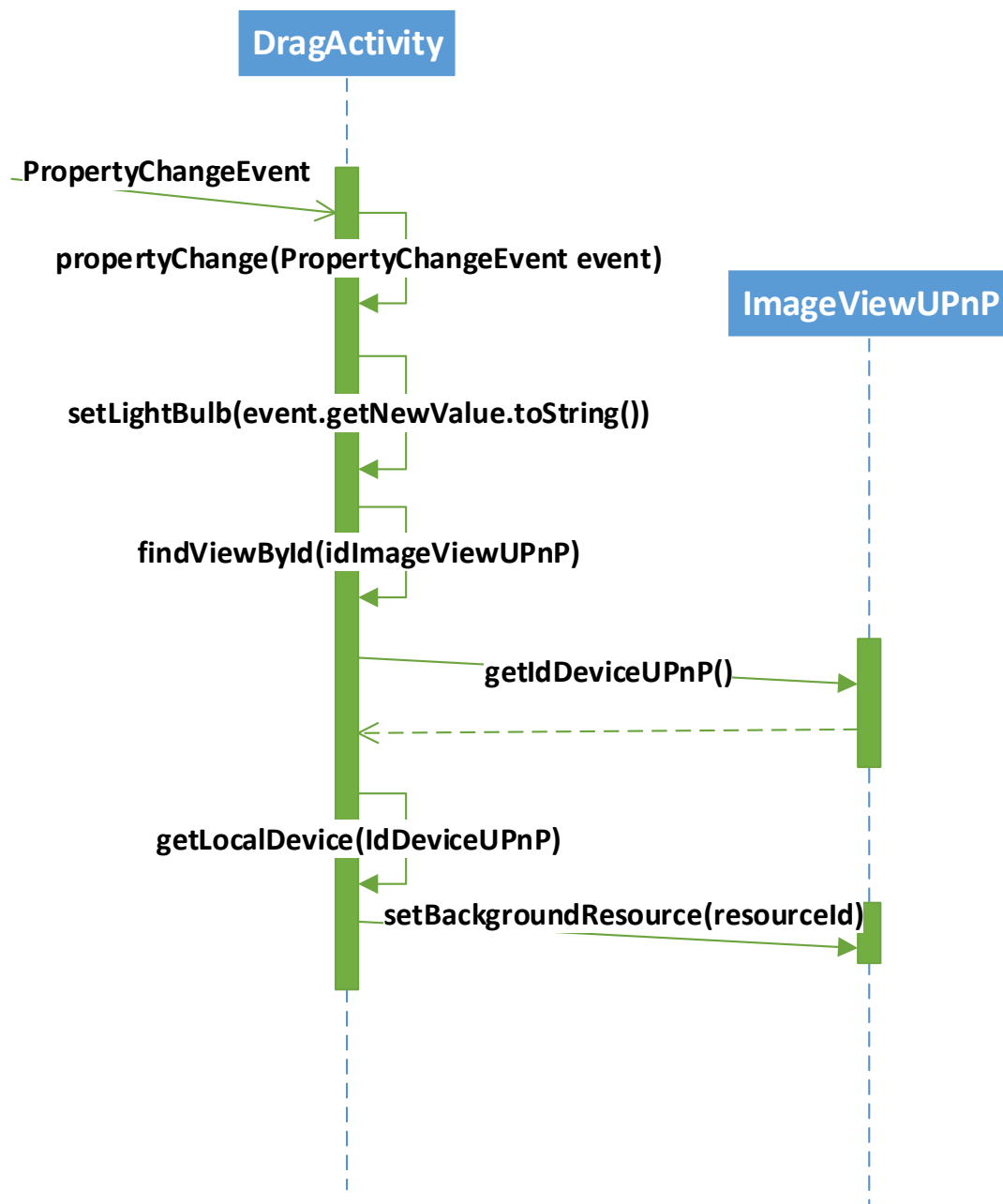


Figura 24 : Diagrama de Secuencia - Captura de Eventos de un Dispositivo UPnP

## 5.7. Diagramas de Paquetes y Clases

La Figura 25 muestra el diagrama de paquetes, que muestra los utilizados en la aplicación Android construida. Este tipo de diagrama es utilizado en la fase de diseño y análisis. Ayudando a crear un diseño conceptual de la información que se maneja.

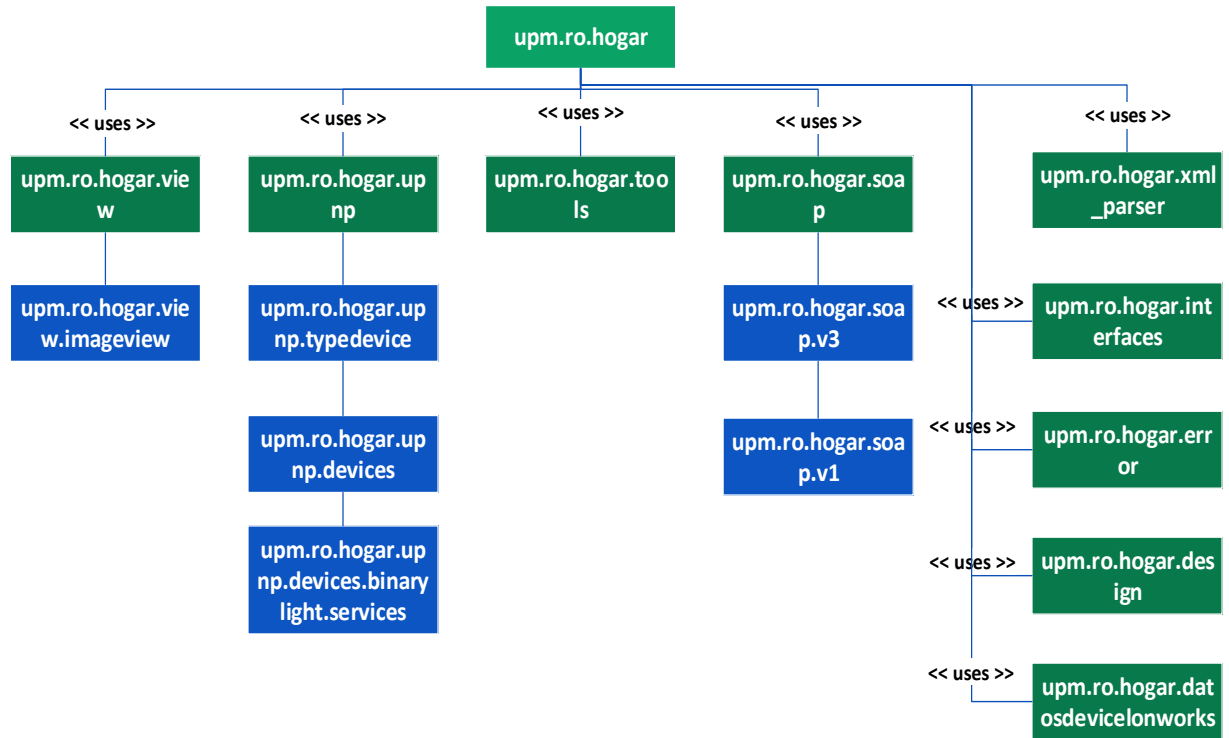


Figura 25 : Diagrama de Paquetes

En la Figura 26 se representan las clases y relaciones existentes en el paquete principal de la aplicación Android que contiene la actividad principal de la Aplicación *MainActivity*. El servicio *MainService* es el servicio para la UPnP Stack y un registro de eventos de los sensores que se almacenara en *RegistryEventAreas*.

upm.ro.hogar



Figura 26 : Diagrama de Clases upm.ro.hogar

## 6. Pruebas y validación

---

Antes de comenzar con el desarrollo fue necesario descargar e instalar el SDK Android en el PC [23]. A continuación, instalar el Plugin Android correspondiente para el entorno de desarrollo Eclipse [16]. Después de tener preparado las herramientas de desarrollo Android, fue necesario importar las librerías Cling *cling-core.jar* y *teleal-common.jar*. Además, de la librería *ksoap2.jar* para el intercambio de mensajes SOAP del Sistema.

A partir de aquí, se continuó con el desarrollo de las clases Java y de los componentes visuales en Eclipse. Una vez concluido el desarrollo se realizaron las pruebas que se describen en los siguientes apartados.

### 6.1. Pruebas de la Aplicación Android con los Paneles Demostrativos

Se ha tomado una serie de imágenes que ayudaran a demostrar el correcto funcionamiento de la Aplicación sobre el entorno domótico de los *Paneles Demostrativos*.

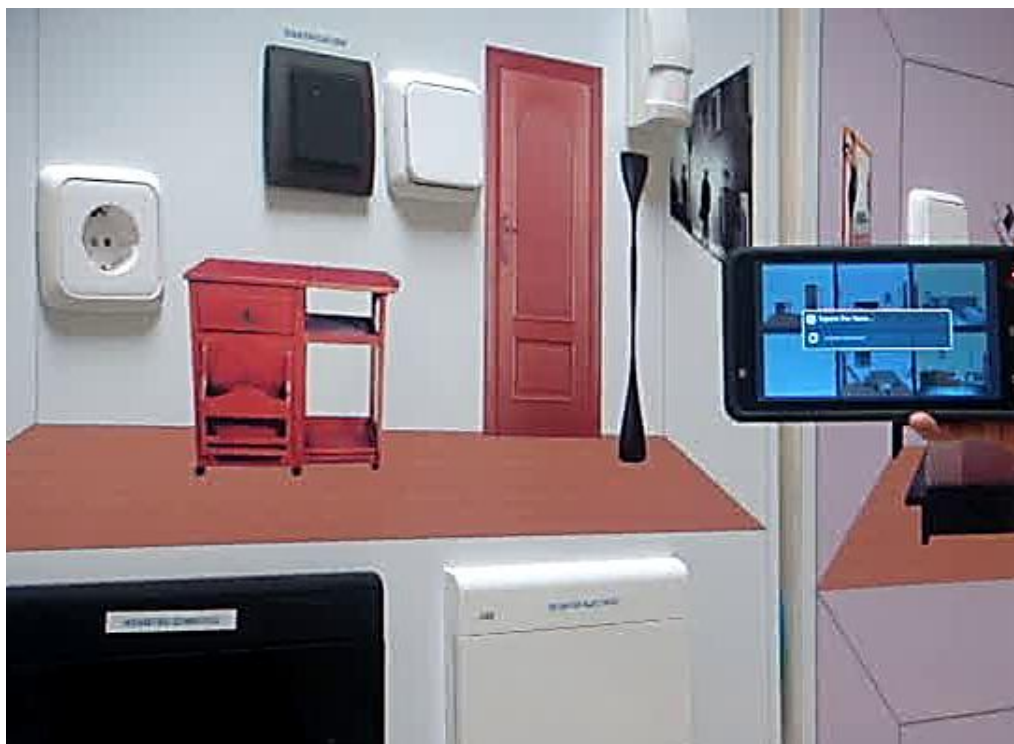


Figura 27 : Cargando Dispositivos en Tablet con Sistema Operativo Android versión 2.2



Figura 28 : Comparación del Área Cocina en Aplicación Android y en los Paneles Demostrativos.

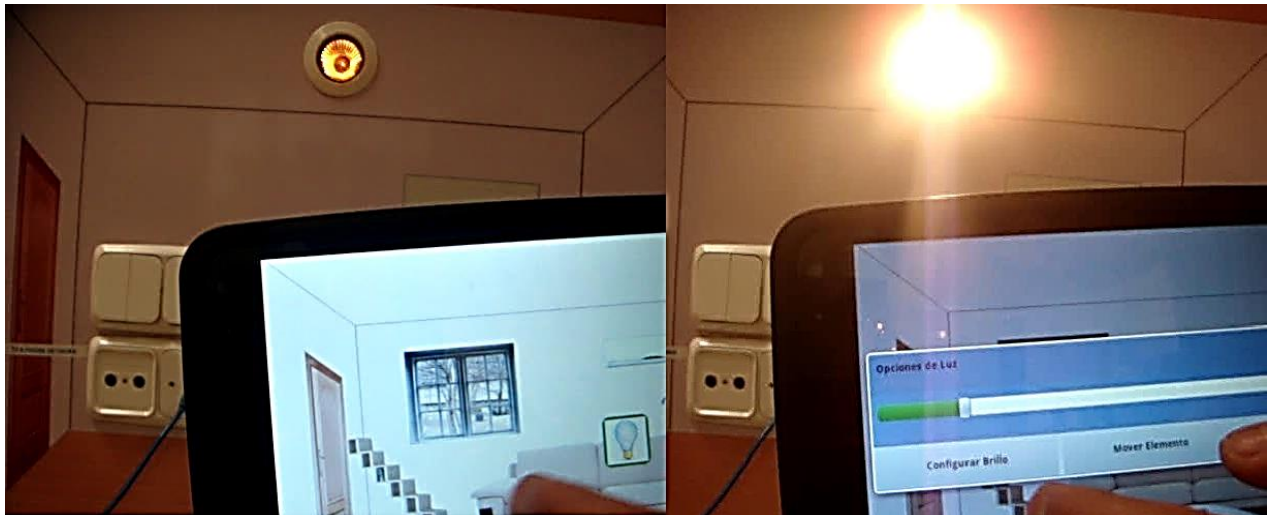


Figura 29 : Prueba de Regulación de Luz

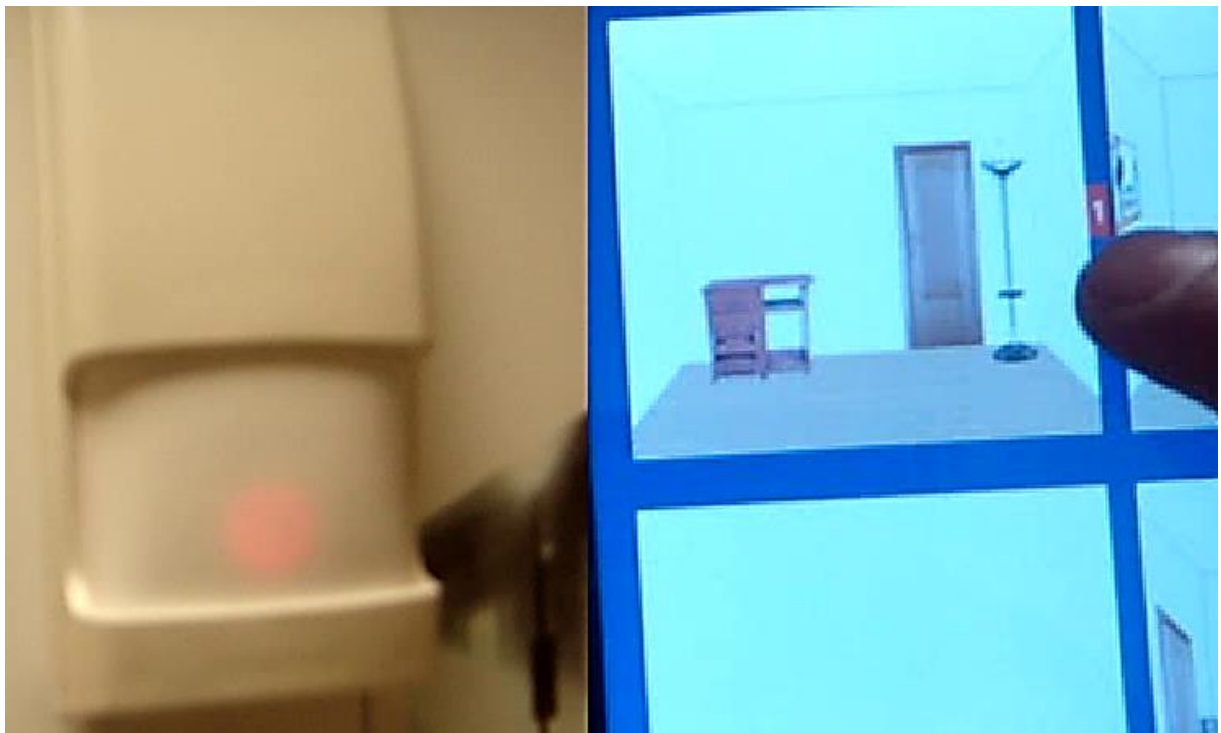


Figura 30 : Prueba de Detención de Sensor de Presencia.





## 7. Conclusiones

---

Con la realización de este proyecto se ha pretendido básicamente dar solución al problema de integrar la tecnología LonWorks con UPnP. Como personalmente tenía cierto interés por el SO Android, se optó por crear un sistema que cumpliera los objetivos marcados integrado en una aplicación Android.

La aplicación Android implementada cuenta con todas las ventajas que ofrece la tecnología UPnP, como por ejemplo: el descubrimiento automático de nuevos dispositivos y su configuración en la red. Se han integrado las tecnologías LonWorks y UPnP en un intento de abordar de forma novedosa la gestión de sistemas LonWorks desde Android.

En otras palabras, la tecnología LonWorks, tras la realización de este PFC, cuenta con una funcionalidad añadida, mejorando sus prestaciones en los entornos en los que sea instalada, pudiendo tratar los dispositivos existentes en el Hogar como dispositivos UPnP.

Para realizar el recubrimiento de los dispositivos LonWorks se han usado con éxito las librerías *Cling*, con las que se ha conseguido crear la red UPnP, que contiene un dispositivo *UPnP virtual* por cada elemento LonWorks existente en los paneles.

La construcción de la aplicación Android se ha hecho con el uso del *Android SDK* para implementar los componentes Android usando para este PFC la versión 2.3 Gingerbread del SO, funcionando además sin problemas en versiones superiores.

Finalmente, los objetivos planteados en este Proyecto, expuestos en el apartado 1.2 se han cumplido con un gran grado de satisfacción.

## 7.1. Trabajos Futuros

En este apartado se enumeran las propuestas que se plantean para futuras líneas de trabajo a seguir a partir de este PFC.

### 7.1.1. Servicios Dinámicos de un Dispositivo UPnP

El control de los dispositivos remotos se plantea como una línea a seguir, siendo para ello necesario crear servicios dinámicos a partir de la información que se recoge en el documento de *Descripción de un Dispositivo UPnP*. Con ello se conseguiría que el propio sistema sea capaz de crear los servicios necesarios o en su caso exportar dichos servicios.

### 7.1.2. Mejorar la vista de la Aplicación Android

La vista de la aplicación Android está basada en la versión 2.3 Gingerbread del SO Android. La vista tendría que refinarse para conseguir una correcta usabilidad en caso de que la aplicación contenga dispositivos UPnP con elementos con estados más complejos.

Por ejemplo, estados que contengan valores no primitivos con información compleja que necesite de un proceso previo de *parseo* o transformación para su posterior uso y control.

### 7.1.3. Acceso a la Aplicación

En este proyecto no se ha contemplado como objetivo del mismo dotar a la aplicación Android de mecanismos de autenticación para su uso. Por lo tanto, Se puede dotar a la aplicación de algún mecanismo que cumpla este servicio.

#### 7.1.4. Compatibilidad de Dispositivos no UPnP

A lo largo de esta memoria se ha hablado de Dispositivos UPnP. El sistema que se ha creado no contempla, el tratamiento de dispositivos no compatibles con la descripción de un dispositivo UPnP, a excepción de los elementos de los dispositivos LonWorks. Sería por tanto interesante crear una línea para incluir en la aplicación dispositivos del Hogar Digital que estén basados en otros estándares, como por ejemplo X.10, Konnex...



## 8. Anexo A: Gestión del Proyecto

En este Anexo se estudia el uso de la descripción de servicios de WSDL describiendo los tipos de campos que tienen los mensajes SOAP. Además se ve qué uso se le da al fichero *Android Manifest* en las aplicaciones Android.

### 8.1. Fichero WSDL Versión 1.1

Un fichero WSDL [24] sirve para describir Servicios Web recomendado por W3C. El fichero WSDL describe la interfaz pública de los servicios, actualmente se encuentra en la versión 2.0 basado en el formato XML. En WSDL se declaran las operaciones y mensajes que describen la forma de comunicación que ofrece el Servicio Web.

En la Figura 31 y Figura 32 ilustran algunos campos importantes que se usan en el intercambio de mensajes SOAP. Algunos de ellos son los campos *UCPTpointName* y *UCPTindex* que son usados por el sistema para identificar unívocamente los dispositivos UPnP creados a partir de los elementos LonWorks.

```
<types>
  <s:schema elementFormDefault="qualified"
    targetNamespace="http://wsdl.echelon.com/web_services_ns/ilon100/v1.1/type">
    <s:complexType name="DataPointReadReturnType">
      <s:sequence>
        <s:element name="UCPTpointName" type="s:string" />
        <s:element name="UCPTfieldName" type="s:string" minOccurs="0" />
        <s:element name="UCPTlocation" type="s:string" />
        <s:element name="UCPTpointUpdateTime" type="s:dateTime" />
        <s:element name="UCPTvalue" type="s:string" />
        <s:element name="UCPTvalueDef" type="s:string" minOccurs="0" />
        <s:element name="UCPTunit" type="s:string" />
        <s:element name="UCPTpointStatus" type="s:string" />
        <s:element name="UCPTpriority" type="s:int" />
        <s:element name="UCPTformatDescription" type="s:string" />
        <s:element name="UCPTbaseType" type="s:string" />
      </s:sequence>
    </s:complexType>
    <s:complexType name="DataPointWriteReturnType">
      <s:sequence>
        <s:element name="UCPTpointName" type="s:string" />
        <s:element name="UCPTfieldName" type="s:string" minOccurs="0" />
      </s:sequence>
    </s:complexType>
    ....
  </s:schema>
</types>
```

Figura 31 : Fichero WSDL parte 1

En la Figura 32 representan algunos tipos de operaciones y mensajes a modo de ejemplo. El fichero consta de un mayor número de operaciones y tipos de mensajes, pero aquí se incluyen únicamente los necesarios para el intercambio de mensajes SOAP.

```
<message name="DataServerWrite">
  <part name="Data" type="s:string" />
</message>
<message name="DataServerWriteResponse">
  <part name="Result" type="s:string" />
</message>

<binding name="MainSoapBinding" type="tns:MainSoapPort">
  ....
  <operation name="DataServerWrite">
    <soap:operation soapAction="http://wsdl.echelon.com/web_services_ns/ilon100/v1.1/action/DataServerWrite" />
    <input>
      <soap:header message="tns:ILON100SoapHeader" part="messagePropertiesPart" use="encoded"
        namespace="http://wsdl.echelon.com/web_services_ns/ilon100/v1.1/message/"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      <soap:body use="encoded" namespace="http://wsdl.echelon.com/web_services_ns/ilon100/v1.1/message/"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:header message="tns:ILON100SoapHeader" part="messagePropertiesPart" use="encoded"
        namespace="http://wsdl.echelon.com/web_services_ns/ilon100/v1.1/message/"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      <soap:body use="encoded" namespace="http://wsdl.echelon.com/web_services_ns/ilon100/v1.1/message/"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  ....
</binding>
<service name="ILON100">
  <port name="MainSoapPort" binding="tns:MainSoapBinding">
    <soap:address location="http://localhost/WSDL/ILON100.WSDL" />
  </port>
</service>
```

Figura 32 : Fichero WSDL parte 2

## 8.2. Android Manifest

El fichero Android Manifest [25] sirve para declarar los servicios y actividades que contiene la aplicación Android mostrando las características y permisos del SDK que utiliza la aplicación.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="upm.ro.hogar"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.CHANGE_WIFI_MULTICAST_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <application
        android:allowBackup="false"
        android:icon="@drawable/icon"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity android:configChanges="orientation|keyboardHidden"
            android:screenOrientation="landscape"
            android:name=".MainActivity"
            android:label="@string/title_activity_main" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="upm.ro.hogar.view.DragActivity"
            android:screenOrientation="landscape"/>
        <service android:name=".MainService"/>
    </application>
</manifest>
```

Figura 33 : Android Manifest

En la Figura 33 se muestra la información que contiene el fichero *AndroidManifest.xml* de la aplicación Android desarrollada. Como se puede ver consta de un servicio denominado *MainService* y de las actividades *MainActivity* y *DragActivity*.





## 9. Anexo B: Manual del desarrollador

---

En este Anexo se describen en primer lugar los requisitos físicos para la puesta en marcha de la aplicación Android.

A continuación se analiza el uso que se le da a las interfaces creadas en la aplicación Android, que ayudaran a un desarrollador a estudiar la lógica de la aplicación.

### 9.1. Instalación del Sistema

El Sistema consta de dos partes físicas que son necesarias para su correcto funcionamiento: una es la utilización de un dispositivo móvil con el S.O Android versión 2.2 Froyo como mínimo y la otra es que el hogar Digital este dotado de la tecnología LonWorks.

Para llevar a cabo la instalación del este sistema, es necesario tener un escenario con las funcionalidades que ofrecen los paneles demostrativos, apartado 2.3.

Por otro lado, el dispositivo móvil debe estar basado en el SO Android versión 2.2 como mínimo denominada Froyo aunque todas las pruebas de validación se han realizado en la versión 2.3 Gingerbread.

Su puesta en marcha es tan simple como instalar en el SO Android la aplicación *HogarDigital* con extensión *".apk"*. La extensión *apk* es la que tienen las aplicaciones en Android.

### 9.2. Interfaces de la Aplicación

Una interfaz es una clase abstracta pura donde todos los métodos son abstractos. Permitiendo a un diseñador establecer el contenido de una clase. Además puede contener dato miembro que son siempre Static y Final.

Los ficheros que contienen las interfaces de la aplicación se almacenan en el paquete *upm.ro.hogar.interfaces* y son los siguientes:

#### ◆ *BuildDeviceLonworks*

Esta interfaz contiene atributos de tipo *String* que son *static* y *final* su contenido son todos los campos posibles para el intercambio de mensajes SOAP, estos campos se pueden ver en la Figura 31.

```
package upm.ro.hogar.interfaces;

public interface BuildDeviceLonworks {
    public static final String UCPT_INDEX = "UCPTindex";
    public static final String UCPT_POINTNAME = "UCPTpointName";
    public static final String UCPT_LOCATION = "UCPTlocation";
    public static final String UCPT_DESCRIPTION = "UCPTdescription";
    public static final String UCPT_FORMATDESCRIPTION = "UCPTformatDescription";
    public static final String UCPT_SIZE = "UCPTdpSize";
    public static final String UCPT_BASETYPE = "UCPTbaseType";
    public static final String UCPT_DIRECTION = "UCPTdirection";
    public static final String UCPT_VALUEDEF = "UCPTvalueDef";
}
```

Figura 34 : Interfaz BuildDeviceLonWorks

#### ◆ *TypesDevices*

En esta interfaz se declara los métodos para cada tipo conocido. Un tipo conocido es básicamente un elemento LonWorks de un dispositivo LonWorks, el cual se puede transformar en un dispositivo UPnP.

El sistema hará uso de esta interfaz a través de la clase *UPnPFiltrar*, que será la encargada de buscar los elementos declarados en los ficheros de configuración.

La clase *UPnPFiltrar* será utilizada para serializar los tipos conocidos sin conocer los atributos que los diferencian, siendo esta clase, la clase padre de todos los tipos conocidos.

```
package upm.ro.hogar.interfaces;

import upm.ro.hogar.upnp.typedevice.TypeUPnPBinaryLight;

* Todos los metodos Get que me devolveran un tipo. Por tanto,
public interface TypesDevices {
    /**
     * Este metodo se encarga de leer el fichero xml
     * correspondiente a esta clase que se debe encontrar
     * en el siguiente path : raw/confighouse.xml
     * que se encuentra
     * @return TypeUPnPBinaryLight
     */
    public TypeUPnPBinaryLight getBinaryLight();

    /**
     * Con este metodo definire el tipo de Dispositivo UPnP que es
     *
     * @param type
     */
    public void setBinaryLight(String type);
}
```

Figura 35 : Interfaz TypesDevices

#### ◆ TypeUPnP

Esta interfaz será la que obligatoriamente tengan que implementar cada tipo de objeto conocido que se quiera agregar a la Red UPnP, dado que esta interfaz consta de los métodos necesarios para poder buscar y formar un nuevo dispositivo en la Red UPnP.

Cada tipo conocido contendrá una lista con los elementos que son de ese tipo. Por tanto, cuando el sistema busque posibles elementos compatibles con UPnP. Buscará en esta lista si hay alguna coincidencia con el elemento que se quiera cargar en la Red UPnP.

```
package upm.ro.hogar.interfaces;

import java.util.ArrayList;

public interface TypeUPnP{

    * Me devolvera el Id de referencia
    public static int id=0;

    public boolean buscarPorPointName(String pointname);

    public String getTypeDevices();

    * @return
    public ArrayList<Integer> getIDicon(String pointname);

    * @param posicion
    public int getIDicon(String pointname,int posicion);

    public ArrayList<DeviceBase> getDevices();

}
```

Figura 36 : Interfaz TypeUPnP

#### ◆ ViewDevices

La interfaz ViewDevices es utilizada por la aplicación Android para conocer los identificadores utilizados de las áreas. Además contiene los métodos que devuelven los servicios de un dispositivo UPnP, como se ve en la Figura 37.

Para llevar a cabo la propuesta del apartado 7.1.1 sería necesario implementar los métodos relacionados con los servicios Remotos.

```
public static final int AREA_KITCHEN =1;
public static final int AREA_LIVINGROOM =2;
public static final int AREA_BEDROOM =3;
public static final int AREA_HALL =4;
public static final int AREA_BATHROOM =5;
public static final int AREA_OTHERS =6;

public static final String BATHROOM = "bathroom" ;
public static final String BEDROOM = "bedroom";
public static final String LIVINGROOM = "livingroom" ;
public static final String KITCHEN = "kitchen" ;
public static final String HALL = "hall" ;
public static final String OTHERS = "others" ;

* Description:
public LocalService getLocalService(String idDevice);
public LocalService getLocalService(String idDevice,String typeService);
public LocalService getLocalService(String idDevice,String typeService,int typeServiceVersion).

* Description:
public RemoteService getRemoteService(String idDevice);
```

Figura 37 : Layout dormitorio.xml

# 10.Anexo C: Conexionado detallado de Red LonWorks

En este Anexo se describen los dispositivos de las áreas de los Paneles Demostrativos utilizados en este PFC. Además se expone el conexionado que se ha hecho con las variables de red nvo y nvi usado en la configuración LonWorks.

## 10.1. Nodo de Control Estándar INS-231

El nodo de control estándar INS-231 es utilizado para conocer cuando se realiza una actuación sobre sus entradas e informar sobre sus estados en ellos se pueden conectar elementos como luces, sensores, actuadores, etc.



FUENTE: PFC TECNOLOGÍA APLICADA AL HOGAR DIGITAL

Figura 38 : Nodo de Control INS-231

Existen un mayor número de variables de las que aquí se va a señalar. Pero aquí solo se ve las variables que han sido usadas para la configuración LonWorks con la aplicación Lonmaker.

TIPO	DEFINICION	DESCRIPCION
<b>SNVT_switch</b>	nvoEstCir1	Esta variable se utiliza para tener conocimiento del estado del Circuito 1
<b>SNVT_switch</b>	nvoEnt3y4	Esta variable se utiliza para enviar por la red el estado de la entrada 3,4 100.0 1 cortocircuito 0.0 0 circuito abierto
<b>SNVT_switch</b>	nvoEnt5y6	Esta variable se utiliza para enviar por la red el estado de la entrada 5,6 100.0 1 cortocircuito 0.0 0 circuito abierto
<b>SNVT_switch</b>	nvoEnt7y8	Esta variable se utiliza para enviar por la red el estado de la entrada 7,8 100.0 1 cortocircuito 0.0 0 circuito abierto
<b>SNVT_switch</b>	nvoEnt9y10	Esta variable se utiliza para enviar por la red el estado de la entrada 9,10 100.0 1 cortocircuito 0.0 0 circuito abierto
<b>SNVT_switch</b>	nviEstLuz1	Esta variable se utiliza para actuar directamente sobre el estado del circuito de salida R0, OR1. Es decir lo que haya conectado en el Circuito 1
<b>SNVT_switch</b>	nviEstLuz2	Es la variable correspondiente para el Circuito 2.
<b>SNVT_switch</b>	nviPulRem1	Esta variable se utiliza para recibir por la Red una pulsación de un nodo Remoto para el Circuito1
<b>SNVT_switch</b>	nviPulRem2	Es la variable correspondiente para el Circuito 2.

Tabla 1 : Variables utilizadas del INS-231

Configuración de los Paneles Demostrativos según el cableado correspondiente a la Figura 2 y que han sido relevantes.

	INS-231 Cocina	INS-231 Baño	INS-231 Entrada	INS-231 Dormitorio
Circuito 1	Actuador de Agua	Luz de Baño	Luz de Entrada	Luz de Dormitorio
Circuito 2	Actuador de Gas			
Entradas 3 y 4		Sensor de Agua		
Entradas 5 y 6	Sensor de Ventana	Sensor de Presencia	Sensor de Presencia	
Entradas 7 y 8	Sensor de Gas			
Entradas 9 y 10	Sensor de Fuego			

Tabla 2 : Relación de conexiones con el Nodo de Control estándar NS-231

## 10.2. Nodo Regulador Analógico INS-360

El nodo regulador analógico INS-360 consta de pulsadores con los que un usuario puede actuar sobre el circuito de iluminación asociado. Además se puede configurar escenas, grabarlas y regular la intensidad de iluminación de las luces que estén conectadas a los circuitos.

Se ha hecho uso de la funcionalidad de regular la intensidad de iluminación para regular la luz del salón a través de la aplicación Android.





Figura 39: Nodo de Control INS-360

El dispositivo INS-360 consta de 6 entradas de conexión, 6 salidas de corte de fase y 3 salidas de 0-10V.

#### ◆ Entradas de Conexión

Los *Paneles Demostrativos* solo tienen conectadas las Entradas E1, E4 y E5 que corresponden con los pulsadores del Circuito 1, Escena 1 y Escena 2 respectivamente.

#### ◆ Salidas de Corte de Fase

En este caso las Salidas de Corte de Fase que se encuentran conectadas son las salidas 5 y 6 que corresponden a las salidas para el Circuito 1 denominado CC1.

#### ◆ Salidas de 0-10V

Para las salidas de tensión Continua solo está conectada la SAL1 como era obvio que es la correspondiente al Circuito 1.

A continuación se indica la definición de algunas variables de Red que son usadas para regular la luz del Salón. Estas variables son necesarias porque la luz del Salón está conectada al Circuito1, aunque existen las mismas variables para el resto de circuitos.

◆ **nvilLumiC1**

Esta variable indica el tanto por ciento de iluminación del Circuito número 1. Para ello es conecta la variable *nvoValorCir1* del nodo origen con la variable *nvilLumiC1* del nodo remoto.

◆ **nvoValorCir1**

Esta variable se utiliza para enviar por la red el tanto por ciento de iluminación del Circuito 1 que tendrá la variable *nvilLumiC1* del nodo origen a la que esté conectada.

### 10.3. Configuración bloque funcional ILON.SNVT\_switch

	Luces	Sensores
<b>nvilInput40</b>		Sensor de Presencia de Entrada
<b>nvilInput39</b>		Sensor de Presencia del Baño
<b>nvilInput37</b>		Sensor de Gas de la Cocina
<b>nvilInput36</b>		Sensor de Agua del Baño
<b>nvilInput35</b>		Sensor de la Ventana de la Cocina
<b>nvoOutput01</b>	Luz del Entrada	
<b>nvoOutput02</b>	Luz del Dormitorio	
<b>nvoOutput03</b>	Luz del Baño	
<b>nvoOutput25</b>	Luz del Salón	

Tabla 3 : Conexiones BF iLon.SNVT\_switch del Dispositivo iLON100



# 11. Anexo D: Manual de Usuario

En este Anexo se expone la interfaz de la Aplicación, la cual se divide en tres vistas, encargándose cada una de mostrar un proceso del sistema.

La primera vista se encarga de mostrar el arranque e inicio de los servicios de la *UPnP Stack*. La segunda se encarga de crear los dispositivos UPnP analizando los elementos de los dispositivos LonWorks definidos en los ficheros de configuración.

Por último, en la tercera vista se describe los dispositivos existentes en un área específica del Hogar.

## 11.1. Inicio de los Servicios de la UPnP Stack

La aplicación empieza con la Figura 40 donde se visualiza la carga de los servicios UPnP encargados de proporcionar a la aplicación Android la UPnP Stack.

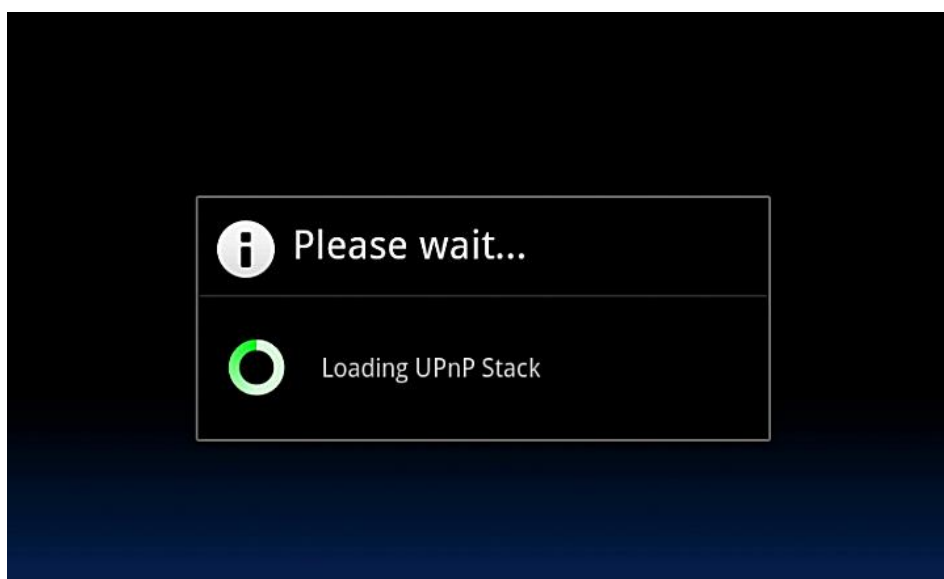


Figura 40 : Cargar Servicios UPnP

En la Figura 41 se ve el resultado final de la vista después de completar el primer procedimiento. En este punto la aplicación ofrece todos los servicios y funcionalidades que ofrece la tecnología UPnP.



Figura 41 : Primera vista

## 11.2. Búsqueda de elementos LonWorks y creación de nuevos dispositivos UPnP.

Para pasar a la segunda vista es necesario hacer *click* en la pantalla en la Figura 41. La Figura 42 muestra el aspecto de la segunda vista que muestra el procedimiento de carga de los elementos LonWorks y la creación de los nuevos Dispositivos UPnP.

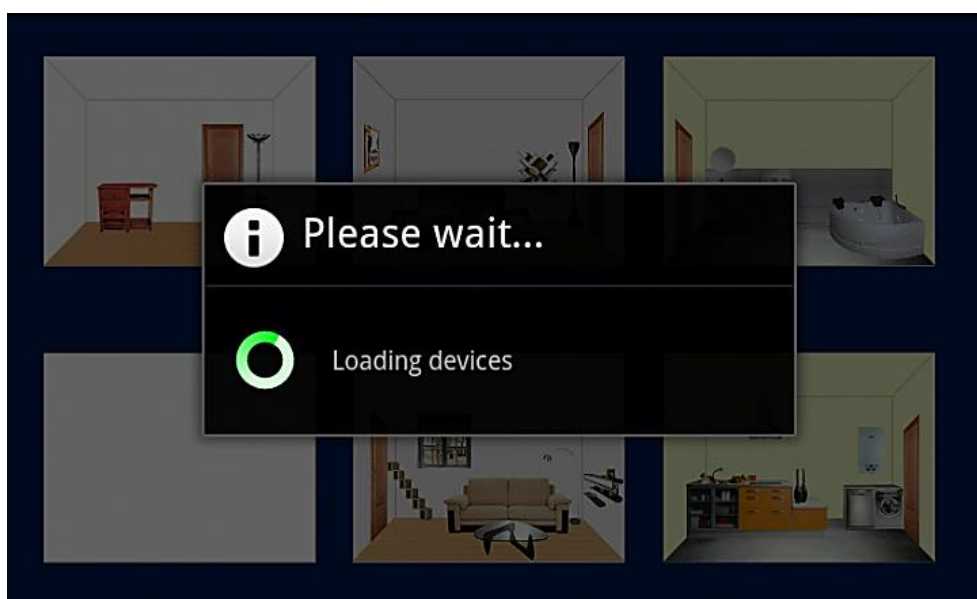


Figura 42 : Buscando elementos LonWorks

Cuando el procedimiento termina de cargarse se puede ver que hay 6 zonas a imagen y semejanza de los *Paneles Demostrativos*, con la salvedad de que existe una Zona denominada “Otros” que muestra los dispositivos UPnP remotos o cualquier otro dispositivo que no haya sido podido ser clasificado según un área específica.



Figura 43 : Segunda Vista

Cuando se selecciona una zona o área de la segunda vista (Figura 43), la aplicación carga los dispositivos UPnP existentes en el área.

La aplicación carga una imagen por defecto cuando sea un dispositivo remoto.

### 11.3. Control y Visualización de Dispositivo UPnP

En la Figura 44 se muestra la tercera vista. En ella se ve cómo se visualizan los dispositivos UPnP.

En la tercera vista cada dispositivo UPnP se representa con una imagen alusiva. Además cuando se altera el estado del dispositivo, cambia el color del borde del componente.



Figura 44 : Tercera Vista

A continuación se incluyen otras imágenes que muestran el comportamiento que puede tener la tercera vista.

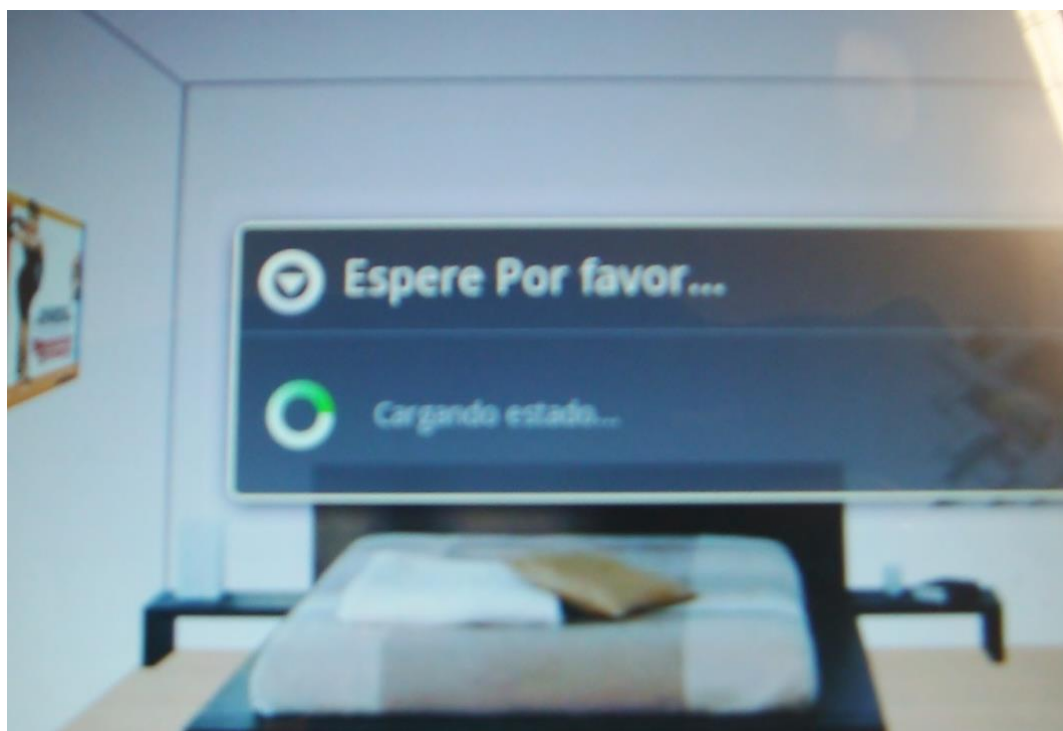


Figura 45 : Proceso inicial de Carga de Estados de los Dispositivos UPnP

En la Figura 45 se ilustra el proceso de cambio de estado de apagado a encendido o viceversa de una bombilla cuando se hace *click* a ese elemento. Además en la Figura 46 se muestra el dialogo para regular la intensidad de una bombilla, mostrándose cuando se mantenga pulsado el elemento.

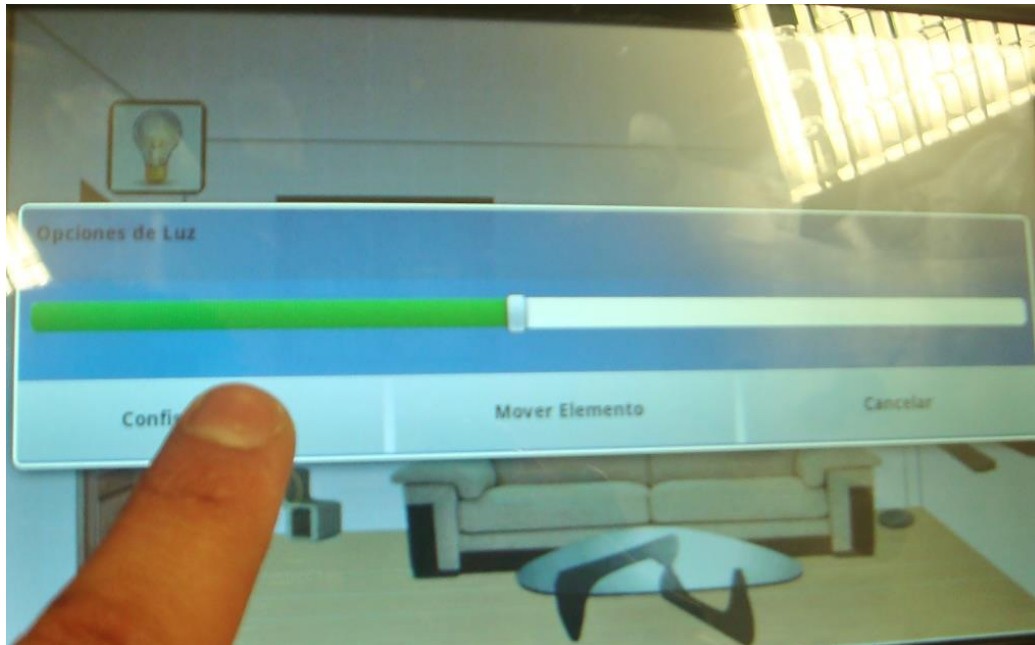


Figura 46 : Dialogo para Regular Luz del Salón.





# 12.Acrónimos

---

<b>API</b>	Application Programming Interface
<b>C2DM</b>	Cloud To Device Messaging Framework
<b>CDMA</b>	Code Division Multiple Access
<b>EDGE</b>	Enhanced Data Rates for GSM Evolution
<b>GSM</b>	Global System Mobile
<b>HMI</b>	Human Machine Interface
<b>IP</b>	Internet Protocol
<b>ISO</b>	International Standards Organization
<b>JVM</b>	Java Virtual Machine
<b>LNS</b>	LonWorks Network Services
<b>LTE</b>	Long Term Evolution
<b>MVC</b>	Model View Controller
<b>OSGI</b>	Open Services Gateway Initiative
<b>PFC</b>	Proyecto Final de Carrera
<b>SDK</b>	Software Development Kit
<b>SNVT</b>	Standard Network Variable Types
<b>SO</b>	Sistema Operativo
<b>SPI</b>	Stateful Packet Inspection
<b>TCP</b>	Transmission Control Protocol
<b>TIC</b>	Tecnologías de la Información y la Comunicación
<b>UML</b>	Unified Modeling Language
<b>UMTS</b>	Universal Mobile Telecommunications System
<b>UPnP</b>	Universal Plug and Play
<b>VGA</b>	Video Graphics Array
<b>W3C</b>	World Wide Web Consortium
<b>WSDL</b>	Web Services Description Language
<b>XML</b>	Extensible Markup Language



# 13. Bibliografía

---

- [1] Adrián Lopez Dominguez and Borja García Oturo, *PFC : Tecnología Aplicada al Hogar Digital.*, Abril 2009, UPM.
- [2] Dlna. (2013) Dlna. [Online]. <http://www.dlna.org/consumer-home/The-Possibilities>
- [3] Casadomo. Casadomo. [Online].  
<http://www.casadomo.com/noticiasDetalle.aspx?c=9&idm=15>
- [4] Rubén de Diego Martinez. Casa Futura. [Online].  
<http://casafutura.diatel.upm.es/rrssmd/documentos/2%C2%AA%20sesion.%20La%20casa%20futura.pdf>
- [5] iLon100 e2 Internet Server Programmer's Reference. (2012, Octubre) Echelon. [Online]. <http://www.echelon.com/support/documentation/archive/manuals/078-0250-01B.pdf>
- [6] UPnP Forum. (2013, Enero) UPnP Forum. [Online]. <http://upnp.org/about/what-is-upnp/>
- [7] Project Cling - 4th Line.org. (2013, Enero) 4th Line.org. [Online].  
<http://4thline.org/projects/cling/>
- [8] 4thline. 4th line.org. [Online]. <http://4thline.org/projects/cling/core/>
- [9] Echelon Corporation. Echelon. [Online].  
<https://www.echelon.com/technology/lonworks/>
- [10] Nelson Dopico. Casa Futura Diatel. [Online].  
<http://casafutura.diatel.upm.es/html/tecs/lonworks.pdf>
- [11] Echelon Corporation. Echelon. [Online].  
<https://www.echelon.com/products/controllers/smartservers/>
- [12] Echelon Corporation. Echelon. [Online].  
<https://www.echelon.com/support/documentation/archive/manuals/078-0289-01A.pdf>
- [13] Echelon Corporation. (2006) Echelon. [Online].  
<http://www.echelon.com/support/documentation/manuals/networktools/078-0333-01A.pdf>

- [14] Jerome DiMarzio, *ANDROID A PROGRAMMERS GUIDE.*: McGraw Hill Professional, 2008.
- [15] Reto Meier, *Android 2 Application Development*. Indianapolis: Wiley Publishing, Enero 2010, ISBN: 978-0-470-56552-0.
- [16] Eclipse Foundation. Eclipse. [Online]. <http://www.eclipse.org/>
- [17] Yu-Hsiang Sheng Wen-Wei Lin Chia-Shou Tu Kuen-Ming Lee. (2006, Febrero) IEE explore. [Online]. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1625956](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1625956)
- [18] Osgi Alliance. Osgi Alliance. [Online]. <http://www.osgi.org/Main/HomePage>
- [19] LonWorks Fundamentals. (2013, Enero) Circon. [Online]. <http://www.circon.com/wp-content/uploads/2011/03/LonWorks-Fundamentals.pdf>
- [20] Broadwin Technology. (2006, Diciembre) broadwin. [Online]. <http://broadwin.com/downloads/pdf/iLON100.pdf>
- [21] Echelon Corporation. (2004) Echelon. [Online]. <https://www.echelon.com/support/documentation/archive/manuals/078-0250-01B.pdf>
- [22] Developers Android. Developers. [Online]. <http://developer.android.com/guide/topics/resources/available-resources.html>
- [23] Developers Android. Get the Android SDK. [Online]. <http://developer.android.com/sdk/index.html>
- [24] Wikipedia Foundation. (2001, Marzo) W3C Web Services Description Language. [Online]. <http://www.w3.org/TR/wsdl>
- [25] Developes Android. The AndroidManifest.xml File. [Online]. <http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [26] Echelon. (2006) Echelon. [Online]. <http://www.echelon.com/support/documentation/manuals/networktools/078-0333-01A.pdf>
- [27] Wikipedia Foundation. (2013, Enero) Wikipedia. [Online]. [http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador)
- [28] Wikipedia Foundation. (2012, Octubre) Wikipedia. [Online]. <http://es.wikipedia.org/wiki/WSDL>

- [29] Developer Resources. (2013, Febrero) ProgressDialog Android. [Online].  
<http://developer.android.com/reference/android/app/ProgressDialog.html>
- [30] Developers Android. (2013, Enero) Dialogs Android. [Online].  
<http://developer.android.com/guide/topics/ui/dialogs.html>